

954724

c-4947

**SUNSPOT—A COMPUTER PROGRAM FOR
PRODUCING OPTIMAL SOLAR SAIL
PLANETOCENTRIC TRAJECTORIES**

by

Lester L. Sackett

September 1977

(NASA-CR-157129) SUNSPOT: A COMPUTER
PROGRAM FOR PRODUCING OPTIMAL SOLAR SAIL
PLANETOCENTRIC TRAJECTORIES (Draper (Charles
Stark) Lab., Inc.) 103 p HC A06/MF A01

N78-24250

CSCL 22A G3/13 20707
Unclas



The Charles Stark Draper Laboratory, Inc.

Cambridge, Massachusetts 02139

Approved for public release; distribution unlimited,

REPRODUCED BY
**NATIONAL TECHNICAL
INFORMATION SERVICE**
U.S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle SUNSPOT—A COMPUTER PROGRAM FOR PRODUCING OPTIMAL SOLAR SAIL PLANETOCENTRIC TRAJECTORIES				5. Report Date September 1977	
				6. Performing Organization Code	
7. Author(s) Lester L. Sackett				8. Performing Organization Report No. c-4947	
9. Performing Organization Name and Address The Charles Stark Draper Laboratory, Inc. Cambridge, Massachusetts 02139				10. Work Unit No.	
				11. Contract or Grant No. NASA 7-100	
12. Sponsoring Agency Name and Address NASA Washington, D.C.				13. Type of Report and Period Covered	
				14. Sponsoring Agency Code	
15. Supplementary Notes Technical Monitor, David B. Smith Jet Propulsion Laboratory, Pasadena, California					
16. Abstract This report contains a description of the input, output and subroutines, including listings, for the SUNSPOT code. SUNSPOT can calculate time optimal planetocentric trajectories including orbit-to-orbit transfer and orbit to a subspace point. Trajectories about the four inner planets can be calculated, and shadowing, oblateness, and solar motion may be included. A penalty function may be included to prevent trajectories which intersect the planet's surface.					
17. Key Words Suggested by Author Solar Sail Optimal Trajectories				18. Distribution Statement Unclassified—Unlimited	
19. Security Classif. (of this report) UNCLASSIFIED	20. Security Classif (of this page) UNCLASSIFIED		21. No. of Pages 103	22. Price	

c-4947

SUNSPOT—A COMPUTER PROGRAM FOR
PRODUCING OPTIMAL SOLAR SAIL
PLANETOCENTRIC TRAJECTORIES

by

Lester L. Sackett

September 1977

Approved: Donald Fraser

The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02139

ACKNOWLEDGMENT

This report was prepared by The Charles Stark Draper Laboratory, Inc. under Subcontract **954724** with the Jet Propulsion Laboratory; **NASA** Contract **NAS 7-100**.

Publication of this report does not constitute approval by the **Jet** Propulsion Laboratory or **NASA** of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
1	INTRODUCTION.....	1
2	INPUT.....	5
3	OUTPUT.....	8
4	COMMON AREAS.....	12
5	SUBROUTINE DESCRIPTION AND LISTINGS.....	19
	CONE.....	20
	CONTL.....	22
	DCROUT.....	24
	DCUBIC.....	27
	DQRTIC.....	29
	EVALMP.....	31
	FCT.....	35
	FUNCT.....	40
	IWPUT.....	44
	ITER.....	53
	OBLATE.....	60
	OUTP.....	63
	OUTPC.....	68
	PLANET.....	71
	PRTM.....	74
	QUAD.....	76
	RK4.....	82
	SHADOW.....	85
	SUN.....	89
	TRAJ.....	92
	VCP.....	96
	LIST OF REFERENCES.....	98

SECTION 1

INTRODUCTION

This report is meant to supplement Draper Laboratory Report R-1113, Optimal Solar Sail Planetocentric Trajectories.^{(1)*} it contains information about the computer code, called SUNSPOT (for SUN-Sail Program for Optimal Trajectories), which was used to generate the results in the report, and should allow one to use this code to produce additional solar sail trajectories. The principal contents are a description of the input and output and brief descriptions of the subroutines and listings of all the subroutines. The code is functionally similar to SECKSPOT, a solar electric orbit transfer program.^(2,3)

The program calculates the solution to a two-point boundary-value problem which arises from the application of optimal control theory. A general flow chart is shown in Figure 1. Figure 2 is a system diagram which illustrates the connection of various subroutines which make up the deck. The subroutines above call those below.

Total size of the object deck compiled on an Amdahl 470 in Fortran H is approximately 59,000 bytes. The size of the individual subroutines is shown in Table 1. A Fortran Scientific Subroutine Package (SSP) routine called DRTNI is also needed, but not shown in the table.⁽⁴⁾

A plotting capability was developed at Draper Laboratory. Data can be written by the output subroutine and stored for later plotting. Since plotting capabilities vary from facility to facility, that function is not discussed in this report.

*

Superscript numerals refer to similarly numbered items in the List of References.

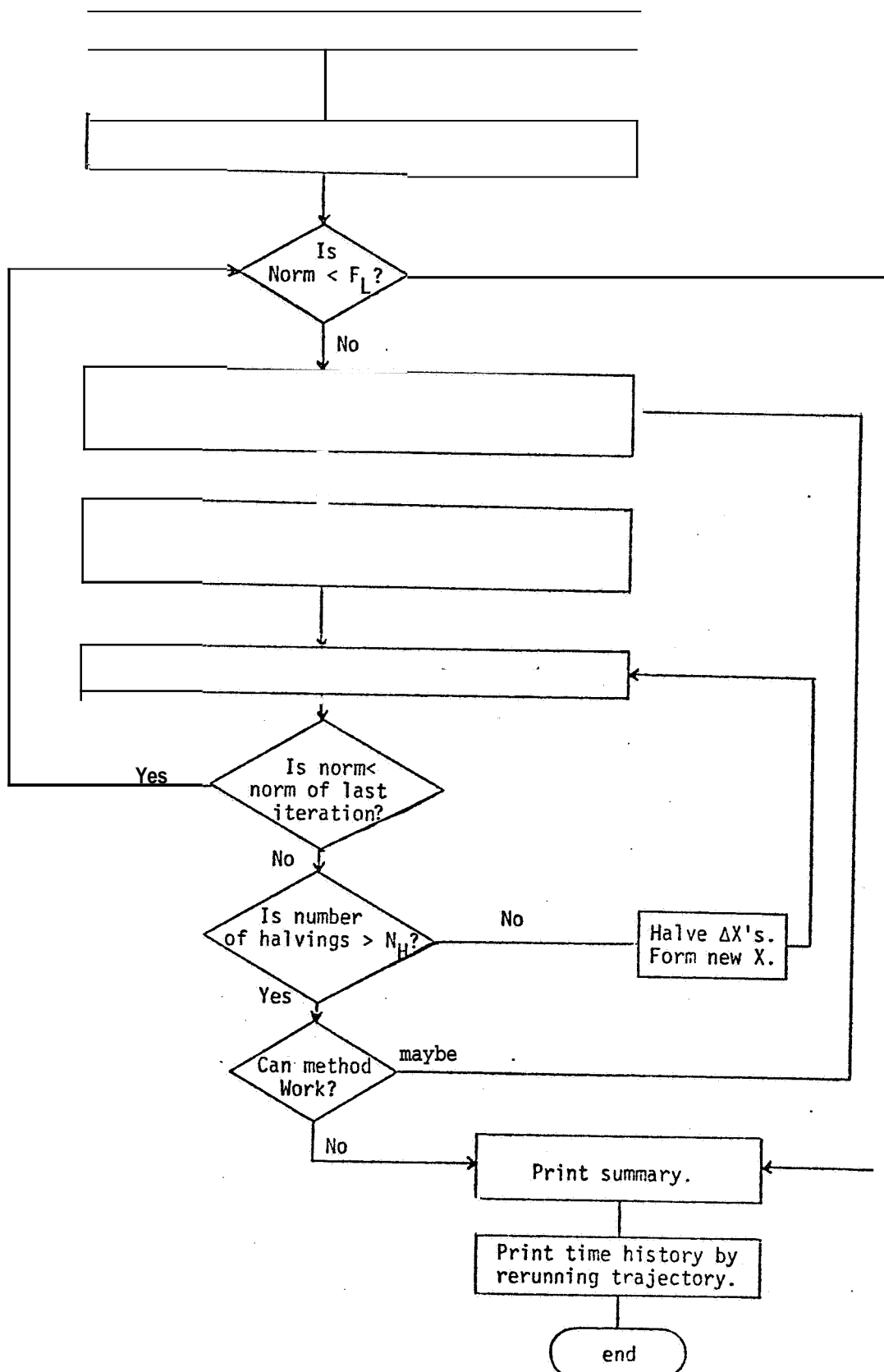


Figure 1. Overall flowchart.

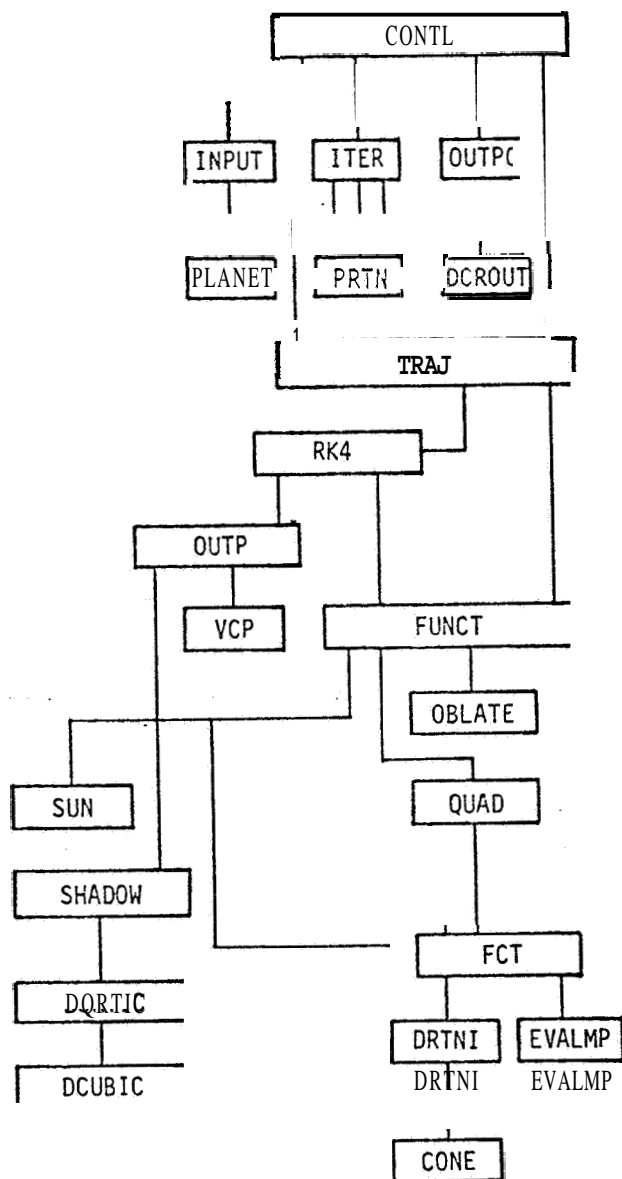


Figure 2. System diagram.

ORIGINAL PAGE IS
OF POOR QUALITY

. Table i. Program subroutine object *deck* size
(thousands of bytes! .

CONE	0.9	OUTP	4.8
CONTL	0.6	OUTPC	2.2
DCUBIC	1.2	PLANET	1.5
DCROUT	2.5	PRTN	0.6
DQRTIC	1.3	QUAD 4	1.0
EVALMP	5.2	QUAD 8	1.2
FCT	3.4	QUAD 16	1.8
FUNCT	2.5	QUAD 32	2.9
INPUT	9.5	RK4	1.7
ITER (NR)	3.2	SHADOW	3.5
ITER (MODNR)	3.9	SUN	1.4
OBLATE	2.0	TRAJ	2.1

SECTION 2

INPUT

The quantities discussed in this section are all read by the sub-program INPUT. Unless otherwise indicated each value is read on a separate line, real variables in fixed format (F25, 15), integer variables beginning with i, j, k, l, m, n are read in format, 12. The inputs for the deck are listed in this section along with a brief description and nominal values, if any.

IPNUM planet flag: 1 = Mercury, 2 = Venus, 3 = Earth, 4 = Mars
 (format, 11)

Initial orbit

W(1) (km) semimajor axis
W(2) eccentricity
W(3) (degrees) inclination
W(4) (degrees) longitude of ascending node
W(5) (degrees) argument of pericenter

Initial guesses

ZL0(1) A , adjoint to semimajor axis
ZL0(2) λ_h , adjoint to orbital element h
ZL0(3) λ_k , adjoint to orbital element k
ZL0(4) λ_p , adjoint to orbital element p
ZL0(5) λ_q , adjoint to orbital element q

The desired final orbit

WF(1) (km) semimajor axis
WF(2) eccentricity, not used if NOP = 3
WF(3) (deg) inclination, not used if NOP = 3

WF(4) (deg) longitude of ascending **node**, not used if NOP = 2 or 3
 WF(5) (deg) argument of pericenter, not used if NOP = 2 or 3
 TF2 (days) guess for final time
 AC (mm/s²) characteristic acceleration.
 TL Julian date at initial time

The following input may be read or, optionally, left at nominal values, IRDFLG is read followed by the addition input or operations and then IRDFLG is read again until IRDFLG = 01 and input is ended.

IRDFLG			NOMINAL
1	End of input		
2	IPR	print flag (for intermediate trajectories)	0
3	NIMAX	maximum number of iterations (if 0, bypass iteration to print time history)	20
4	TFMAX2	(days) maximum flight time	500.
5	DT2	(days) time step for integrator	10.
6	UEB	upper error bound for integrator (not used with RK4)	1.D10
7	EW	error weights for integrator (5D6.1, 2 lines) (not used with RK4)	1.,1,1,1,1,0 ... /
8	UTKM	equatorial radius (km)	Set in PLANET
9	GM	(km ³ /s ²) gravitational constant	Set in PLANET
10	NO?	= 1, five orbital elements specified at TF = 2, three orbital elements specified at TF = 3, semimajor axis specified at TF	1
11	Sets oblateness	AJ2, = 0.D0	Set in PLANET
12	STEP	step size for numerical differentiation in ITER, 6 dim., sixth element for time variation of Hamiltonian	1.D-2
	KSTEP	= 0, step as fraction in ITER = 1, step as constant in ITER (except STEP (6))	1

13	ISON	= 0, shadow effect off	0
		= 1, shadow effect on	
14	ISUN	= 0, sun distance effect on thrust off	0
		= 1, effect on	
15	CD(1)	coefficients in	0.5
	CD(2)	acceleration factor	0.5
	CD(3)		0.0
16	IE	= 1, if equatorial frame, solar motion	1
		= 2, if equatorial frame, no solar motion	
		= 3, if ecliptic-frame, solar motion	
		= 4, if ecliptic frame, no solar motion	
		= 5, if planetary frame, solar motion	
		= 6, if planetary frame, no solar motion	
17	FLIM	norm limit in iteration routine (format, D6.1)	1.D-6
18	XNU	pericenter penalty function weighting factor	0.0
19	NORB	= 0, no orbit print	0
		= 1, ..., 999, orbit print on	
		NORB points of an orbit (format, I3)	
20	Sets flag so that plot data is stored		
21	IQ	orbit divided into IQ separate quadrature intervals (1-10)	2

SECTION 3

OUTPUT

Most of the output is self-explanatory and a look at an example will familiarize the user with it. There are certain basic groups of output. The first is the printing of the read-in initial data and a few internally set constants. Normally this will be followed by output from the iterator. After convergence, a summary of characteristics of the converged trajectory is printed. Finally, a time history of the converged trajectory will be printed. Usually, even if convergence was unsuccessful, a time history of the last trajectory to be calculated will be printed.

The printing of the initial data should be understandable. There are a few abbreviations used.

A	semimajor axis
E	eccentricity
I	inclination
LON ASC NODE	longitude of ascending node
ARG PERIC	argument of pericenter
MS	meters/second
P.R.	planet radii
UTKM	internal units to kilometers
UTS	internal units to seconds
UTD	internal units to days
UTKG	internal units to kilograms
UTKW	internal units to kilowatts
UTMS2	internal units to meters/second ²

After the initial input print, the iteration begins. The iteration number (ITER NO.) and the total number of calls to TRAJ are printed followed by **X**, the iteration parameters (XLO), then **Y**, the error in the final conditions. The final conditions are the final values of **a**, **h**, **k**, **p**, **q**, **H**, if **NOP** = 1. If **NOP** = 2 the final conditions are **a**, **e**, $\tan \frac{i}{2}$, λ_ω , λ_Ω , **H** unless the final eccentricity or inclination is zero. If eccentricity is zero the second condition is **h** and the fourth **k**; if inclination is zero, the third is **p** and the fifth is **q**. If **NOP** = 3, the final conditions are **a**, λ_h , λ_k , λ_p , λ_q , **H**. Then the final time (TF) is printed in internal units, followed by, **FO**, the sum of the squares of the errors in the final conditions. For convergence this value must be less than **FLIM**, the "norm limit in ITER". In order to calculate the partial derivative matrix or sensitivity matrix, the nominal values of "**X**" are changed slightly by inputted amounts; these perturbed values of **X(X(I) + DX(I))** are next printed followed by the corresponding **Y**. Then the partial derivative matrix is printed as well as its determinant. This matrix is inverted and premultiplies the error vector to obtain the changes in the **X's**, **DELX:S**, which are next printed.

A new trajectory is calculated and the sum of the squares of the errors in the final conditions is printed (**F1**). If this is smaller than **FO**, a new iteration begins; if it is larger than **FO**, the **DELX:S** are halved and printed. This continues until **F1** < **FO** or until a certain number of halvings. What follows depends on how well the method converges and on whether the Newton-Raphson or modified Newton-Raphson subprogram is used. Further output is basically permutations of the above, terminating with convergence or a message indicating lack of success.

After exit from the iteration, a summary of characteristics of the last trajectory (the optimal, if convergence was successful) is printed. Included are the actual final orbital elements, the error in the final orbital elements, the values for the iteration parameters, and the final time.

Next is printed a time history of the final (optimal if convergence was successful) trajectory. If **NIMAX** = 00, then a time history is printed immediately following printing of the input data, bypassing the iteration routine, and summary print.

The low thrust trajectory time history at each time step is printed. First is printed **TIME** in various units. The time step number is also printed. Next are printed the equinoctial orbital elements (**a**, **h**, **k**, **p**, **q**) followed by the classical orbital elements (**a**, **e**, **i**, **SL**, **ω**). Next is the

costate, then the state derivative, then the costate derivative and then the value of the Hamiltonian, the period (hours), pericenter and apocenter (km). The characteristic acceleration divided by the sun distance squared, and the value of C_3 . If shadowing is included the time spent in shadow is printed in hours and as a fraction of the period (if the orbit passed through shadow). The entry angle ($^\circ$) and exit angle ($^\circ$) are printed. If NORB was set to a nonzero value then there is additional output.

The optionally printed output for an orbit at each time step includes the sun direction in the equinoctial coordinate frame and the planet-sun distance in units of the planets semimajor axis. Several spacecraft parameters are printed at a number of points on the orbit (the number of points = NORB). The points are at equal units of eccentric longitude. At each point the spacecraft location is printed including the eccentric longitude and the \hat{f} and \hat{g} components X_1 and Y_1 in units of planet radii. The following spacecraft parameters are printed with the spacing indicated here:

PRIMER CONE A	THRUST CONE A	PSIV	ACC FACT	T/W RATIO
UF	UG	UW		

All angles are in degrees. The meaning of these terms follows, Equation numbers refer to Reference 1.

PRIMER CONE A	the primer vector cone angle, β , as in Eq. (3.50)
THRUST CONE A	the force vector cone angle, θ , as in Eq. (3.27)
PSIV	the force vector clock angle, ψ , as in Eq. (3.52), except that ψ has the opposite sign, where the reference vector is the velocity vector
ACC FACTOR	the value of the acceleration factor, $c_1 + c_2 \cos 2\theta + c_3 \cos 4\theta$ as in Eq. (3.26)
T/W RATIO	the ratio of the photon force with the local gravitational force
UF, UG, UW	the \hat{f} , \hat{g} , and \hat{w} components, respectively, of the thrust acceleration unit vector, in the equinoctial coordinate system

A number of error messages are scattered through the code. A few will be mentioned here. Several, in INPUT, call attention to bad input data. For bad input data following an IRDFLG value, a message, IRDFLG = (number), is printed. In some cases additional information is

given. When shadowing is included, a message, **ISHAD = 1**, indicates that only **one** shadow crossing was found. This usually arises from small numerical inaccuracies in solving the quartic equation and can usually be ignored unless a number of orbits have intersected the planet's surface.

SECTION 4

COMMON AREAS

There are several common areas which are shared by several of the subroutines. Table 2 lists these common areas and indicates which subroutines share them. Those subroutines which have no common areas are not included. Following is a list of common areas, the variables in these areas, the definition of variables, and a list of subroutines which contain the particular common areas. Equation numbers refer to Reference 1.

Common area: A/A, AMU, PI

A	Characteristic acceleration divided by semimajor axis of planet squared
AMU	Gravitational constant, μ
PI	π

Shared by:

FCT, FUNCT, INPUT, OUTP, TRAJ

Common area: CCOM/CD(4)

CD(1)	} C_1, C_2, C_3 coefficients in acceleration factor, Eq. (3.3)
CD(2)	
CD(3)	
CD(4)	Cutoff value of θ , Eq. (3.7)

Shared by:

FCT, INPUT, OUTP, OUTPC, CONE

Common area: CON/CB, SB, THETA, CTHETA, STHETA, C2T, C4T, S2T, S4T

CB	Cosine of primer vector cone angle
SB	Sine of primer vector cone angle
THETA	Force vector cone angle
CTHETA	Cosine of force vector cone angle
STHETA	Sine of force vector cone angle

Table 2. Common areas.

SUBROUTINES														
	CONE	CONTL	EVALMP	FCT	FUNCT	INPUT	ITER	OUTP	OUTPC	PLANET	PRTN	SHADOW	SUN	TRAJ
A				X	X	X		X						X
CCOM	X			X		X		X	X					
CON	X			X				X						
DY							X							X
ELEM						X			X					X
F						X	X							
INT		X				X	X	X						X
IS					X	X		X						
JD						X				X				
J2					X	X				X				
NU					X	X								X
ORBIT						X		X						
ORBIT1				X				X						
ORBIT2			X	X				X						
PLNTS						X				X				
PLOT						X		X						
Q					X	X								
SHAD					X			X				X		
SOL				X	X			X				X	X	
T						X	X		X		X			X
TC						X			X					X
TERRA						X				X			X	
TRA						X								X
UNITS						X		X	X	X				
WF						X			X					
XMMM						X	X		X		X			X
Z									X					X

COMMAND AREAS

C2T Cosine of two times the force vector cone angle
 c4T Cosine of four times the force vector cone angle
 S2T Sine of two times the force vector cone angle
 S4T Sine of four times the force vector cone angle

Shared by:

FCT, OUP, CONE

Common area: DY/DYDT(6)

DYDT The partial of the final conditions with respect to time

Shared by :

ITER, TRAJ

Common area: ELEM/ZP0(5), ZPF(5)

ZP0 The initial state

ZPF The desired final equinoctial orbital elements

Shared by:

INPUT, OUTPC, TRAJ

Common area: F/FLIM, KSTEP

FLIM The requirement on the limit of the norm of the errors
 in the iterator.

KSTEP Flag indicates whether STEP refers to a fixed increment
 or a fractional increment

Shared by :

INPUT, ITER

Common area: INT/IPR, IDIM, IDIM2, NIMAX

IPR Print flag used in OUP and OUTHI

IDIM Equal to the dimension of the state plus costate (10)

IDIM2 Equal to dimension of the state (5)

NIMAX The maximum number of iterations of the iterator

Shared by:

CONTL, INPUT, ITER, OUP, TRAJ

Common area: IS/ISUN, ISON

ISUN Flag indicating if solar distance effect is included

ISON Flag indicating if shadowing is included

Shared by :

FUNCT, INPUT, OUP

Common area: JD/TL

TL Julian data at launch

Shared by :

PLANET, INPUT

Common area: J2/AJ2

AJ2 The oblateness coefficient

Shared by :

FUNCT, INPUT, PLANET

Common area: NU/XNU

XNU Weighting constant in penalty function, Eq. (3.73)

Shared by :

FUNCT, INPUT, TRAJ

Common area: ORBIT/NORB

NORB Indicates if there should be orbit print and the number of points printed

Shared by :

INPUT, OUTP

Common area: ORBIT1/VEC(3), PA

VEC The thrust direction vector

PA Acceleration factor, Eq. (3.3)

Shared by:

FCT, OUTP

Common area: ORBIT2/X1, Y1, PR(2, 2), X1DOT, Y1DOT

X1 \hat{r} component of spacecraft location in orbit (Eq. 3.16, Reference 1)

Y1 \hat{g} component of spacecraft location in orbit (Eq. 3.17, Reference 1)

PR(2, 2) Partial derivatives of X1 and Y1 with respect to h and k (Table B-2, Reference 1)

X1DOT \hat{r} component of velocity, Eq. (3.18)

Y1DOT \hat{g} component of velocity, Eq. (3.19)

Shared by :

EVALMP, FCT, OUTP

Common area: PLNTS/IPNUM

IPNUM Integer indicating planet number

Shared by:

INPUT, PLANET

Common area: PLOT/IPL

IPL Flag indicates if data should be written and stored for later plotting

Shared by:

INPUT, OUIP

Common area: Q/IQ

IQ The number of segments into which the orbit is divided, so that the quadrature is called for each segment to calculate the thrust effect.

Shared by:

FUNCT, INPUT

Common area: SHAD/FEN, FEX, DFEN(5), DFEX(5), ISHAD

FEN The eccentric longitude at shadow entry

FEX The eccentric longitude at shadow exit

DFEN(5) The partial of FEN with respect to the equinoctial orbital elements

DFEX(5) The partial of FEX with respect to the equinoctial orbital elements

ISHAD A flag indicating whether or not a particular orbit passed through the planet's shadow

Shared by :

FUNCT, OUTP, SHADOW

Common area: SOL/RSUN(3), RS

RSUN The unit vector pointing toward the sun from the planet

RS The distance from planet to sun divided by semimajor-axis

Shared by :

FCT, FUNCT, OUTP, SHADOW, SUN

Common area: T/TF, TO

TF Flight time

TO The initial time (measured from launch time)

Shared by:

INPUT, ITER, OUTPC, PRTN, TRAJ

Common axea: TC/NOP

NOP Flag indicates whether one or three final or five
 final orbital elements are specified

Shared by :

INPUT, OUTPC, TRAJ

Common area: TERRA/AE, EC, W, ENE, AN, ECLMTX, EQUMTX

AE Planet's semimajor axis (in A.U.)
EC Eccentricity of planet's orbit
W Planet's argument of perihelion
ENE Planet's mean orbital motion
AN The mean anomaly of the planet at the time of launch
ECLMTX Matrix for conversion to ecliptic coordinate frame
EQUMTX Matrix for conversion to equatorial frame

Shared by:

PLANET, SUN, INPUT

Common area: TRA/TFMAX, DT, UEB, EW(10)

TFMAX The maximum time of flight allowed
DT The time step used by the differential equation integrator
UEB The upper error bound (not used by RK4)
EW Error weights (not used by RK4)

Shared by: .

INPUT, TRAJ

Common area: UNITS/UTS, UTM, UTH, UTD, UTKM, DTR, UTKG, UTKW, UIMS?

UTS Internal units to seconds
UTM Internal units to minutes
UTH Internal units to hours
UTD Internal units to days
UTKM Internal units to kilometers
DTR Degrees to radians
UTRG Internal units to kilograms
UTKW Internal units to kilowatts
UIMS2 Internal units to meters/second²

Shared by

PLANET, INPUT, OUTP, OUTPC

Common area: WF/WF(5)

WF The desired final classical orbital elements

Shared by :

INPUT, OUTPC

Common area: XMMM/ZL0(5), STEP(6), ZERF(6)

ZL0 The iteration parameters

STEP Step size used to numerically evaluate partial derivative (or sensitivity) matrix in iterator

ZERF The error in the final conditions

Shared by :

INPUT, ITER, OUTPC, PRTN, TRAJ

Common area: Z/ZF(10), DZ(10)

ZF The state and costate

DZ The derivative of the state and costate

Shared by:

OUTPC, TRAJ

SECTION 5

SUBROUTINE DESCRIPTION AND LISTINGS

This section contains descriptions of all subroutines including input and output, common areas, subroutines which are called or called by. A listing follows the description. A variable in the argument list or the common areas is underlined once if it is output, twice if input or three times if both. Aliases are sometimes given in parentheses following the subroutine name. In addition to the subroutines contained in this section, an IBM Scientific Subroutine, DRTNI is needed. ⁽⁴⁾
Equation numbers refer to the final report, Reference 1.

Subroutine CONE

Description:

Calculates $\frac{\partial H}{\partial \theta}$ and its derivative (see Eq. (3.60) and (3.64)).

Used to calculate cone angle by the Newton method.

Argument List:

THETA, FT, DERFT

THETA Iterative value of thrust cone angle

FT Iterative value of $\frac{\partial H}{\partial \theta}$

DERFT Iterative value of $\frac{\partial^2 H}{\partial \theta^2}$

Common Areas:

CCOM/CD(4)

CON/CB, SB, DUB(7)

Called by:

DRTNI

C	CCHE	00000010
		00000020
2	VERSION OF CORE TO BE USED WITH NEWTON ITERATOR	00000030
C	CALLED BY DRTNI	00000040
C	CALCULATES DH/DTHETA AND DERIVATIVE	00000050
C		00000060
	SUBROUTINE CONE(THETA,FT,DERFT)	00000070
C		00000080
	IMPLICIT REAL*8(A-H,O-Z)	00000090
C		00000100
	COMMON /CCOM/CD(4)	00000110
	COMMON /CON/CB,SB,DUB(7)	00000120
C		00000130
C		00000140
C		00000150
	CTHETA= DCOS(THETA)	00000160
	STHETA= DSIN(THETA)	00000170
	C2T= ECOS(2.D0*THETA)	00000180
	C4T= DCOS(4.D0*THETA)	00000190
	S2T= DSIN(2.D0*THETA)	00000200
	S4T= DSIN(4.D0*THETA)	00000210
C		00000220
	B= CD(1)+CD(2)*C2T+CD(3)*C4T	00000230
	BP= -2.D0*CD(2)*S2T-4.D0*CD(3)*S4T	00000240
	BPP= -4.D0*CD(2)*C2T-16.D0*CD(3)*C4T	00000250
	D= CTHETA*CB+STHETA*SB	00000260
	DP= -STHETA*CB+CTHETA*SB	00000270
	DPP= -D	00000280
C		00000290
	FT= BP*D+B*DP	00000300
	DERFT= 2.D0*BP*DP+BPP*D+B*DPP	00000310
		00000320
	RETURN	00000330
	END	00000340

Subroutine CONTL

Description:

This is the main "driver" program.

Common Areas:

INT/IPR, IDIM, IDIM2, NIMAX

Calls Subroutines:

INPUT, ITER, OUTPC, TRAJ

CCNTL/CONTLSS	00000010
C SOLAR SAIL	00000020
C THIS IS THE MAIN CONTROLLING PROGRAM FOR FINDING THE	00000030
C OPTIMAL TRAJECTORY FOR PLANETOCENTRIC SOLAR SAIL	00000040
C SPIRAL, ESCAPE OR CAPTURE.	00000050
C	00000060
C	00000070
C	00030090
C	00003090
C	00000100
C	00000110
IMPLICIT REAL*8(A-H,O-Z), INTEGER (I-N)	00000120
COMMON /INT/IPR, IDIM, IDIM2, NIMAX	00000130
EXTERNAL TRAJ, PRTH	00000140
C	03000150
C	00000160
C ALL SETTING OF CONSTANTS AND INITIAL READ AND WRITES	00000170
C ARE DONE BY THE SUEEOUTINES INPUT, PLANETS	00000180
C	00000190
CBLL INPUT	00000200
IF (NIMAX.EQ.0) GC TO 10	00000210
WRITE (6,7001)	00000220
C	00000230
C THE ITERATOR ROUTINE SOLVES THE 2PBVP FOR THE OPTIMAL TRAJECTORY	00000240
C	00000250
CALL ITER(KOUNT, HI, TRAJ, PRTH)	00000260
IF (NI.EQ. 9999) WRITE (6,1002)	00000270
WRITE (6,1003)	00000280
C	00000290
C OUTPC PRINTS A SUMMARY OF THE OPTIYAL TRAJECTORY VALUES	00000300
C	03000310
CALL OUTPC(KOUNT)	00000320
C	00000330
C TIME AISTORY OF THE OPTIMAL TRAJECTORY IS CALCULATED BPD PRINTED	00000340
C	00000350
10 WRITE (6,1004)	00000360
IPR= 1000000	00000370
CALL TRAJ	00000380
C	00000390
IF (NI.EQ.9999) WRITE (6,1002)	00000400
IF (NI.BE.9999) YRITE (6,1005)	00000410
STOP	00000420
1001 PORKAT (18H1 ITERATION BEGINS)	00000430
1002 FORMAT (29H0 OPTIMIZATION NOT SUCCESSFUL)	00000440
1003 FORMAT (43H1 CONVERGED VALUES FOR OPTIMIZED TRAJECTORY)	00000450
1004 PORKAT (36H1 TIME HISTORY OF OPTIMAL TRAJECTORY)	00000460
1005 FORMAT (30H0 PROGRAM HAS RUN SUCCESSFULLY)	00000470
END	00000480

ORIGINAL PAGE IS
OF POOR QUALITY

Subroutine DCROUT (DCROUT10)

Description:

This has a square matrix and a vector as input. It inverts the matrix and premultiplies the vector by the inverse.

Argument List:

AA, R, D, EPS, NI, M, IND

AA	The matrix which is to be inverted
R	As input, the vector to be multiplied by the inverted matrix, as output, the resultant vector
D	The determinant of AA
EPS	An input, a small quantity which is used to check for a singular matrix
NI	Dimension of the matrix and vector
M	Flag set to 1 as input
IND	Flag, if $\neq 0$, matrix is singular

Called by :

ITER

```

DCROUT/DCROUT10 6 DIM.
SUEHODTINE DCRUT(AA,R,D,EPS,NI,M,IND)
DOUBLE PRECISION A,R,D,EPS,T,S,P,DT,AA
DIMENSION A(6,6),R(6,1),AA(6,6)
5 IND=0
N=NI
DO 6 I=1,N
DO 6 J=1,N
6 A(I,J)=AA(I,J)
IF(M) 10,25,25
10 M=N
DO 20 I=1,N
DO 15 J=1,N
15 R(I,J)=0.D0
20 R(I,I)=1.D0
25 IC=0
II=0
T=DABS(A(1,1))
DO 35 I=2,N
IF(T-DABS(A(I,1))) 30,35,35
30 II=I
T=DABS(A(I,1))
35 CONTINUE
IF(II) 40,65,40
40 IC=IC+1
IF(M) 45,55,45
45 DO 50 J=1,M
S=R(1,J)
R(1,J)=R(II,J)
50 R(II,J)=S
55 DO 60 J=1,N
S=A(1,J)
A(1,J)=A(II,J)
60 A(II,J)=S
65 P=A(1,1)
IF(DABS(P)-EPS) 70,70,75
70 IND=1
D=0.D0
GO TO 200
75 DO 80 J=2,N
80 A(1,J)=A(1,J)/P
IF(M) 85,95,85
85 DO 90 J=1,M
90 R(1,J)=R(1,J)/P
95 DO 170 K=2,N
KM=K-1
T=-1.D0
DO 105 I=K,N
DO 98 J=1,KM
98 A(I,K)=A(I,K)-A(I,J)*A(J,K)
IF(T-DABS(A(I,K))) 100,105,105
100 T=DABS(A(I,K))
II=I
105 CONTINUE
IF(II-K) 110,135,110
110 IC=IC+1
IF(M) 115,125,115
115 DO 120 J=1,M
S=R(K,J)
R(K,J)=R(II,J)
120 R(II,J)=S

```

```

00000010
00000020
00000030
00000040
00030050
00000060
00390070
000000A0
00000090
00090100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000250
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
03000500
00000510
00000520
00000530
00000540
00000550
09000560
00000570
00000580
00000590
00000600
00000610

```

125 DO 130 J=1,N	00000620
S=A (K,J)	00000630
A (K, J)=A (XI, J)	00000640
130 A (II, J)=S	00000650
135 DT=A (K, K)	00000660
IF (DABS (DT) -EPS) 70, 70, 140	00000670
140 P=P*DT	00000680
IF (K-N) 145, 155, 145	00000690
145 KP=K+1	00000700
DO 150 J=KP, N	00000710
DO 148 I=1, KM	00000720
148 A (K, J)=A (K, J) -A (K, I) *A (I, J)	00000730
150 A (K, J)=A (K, J)/DT	00000740
155 IF (M) 160, 170, 160	00000750
160 DO 165 J=1, M	00000760
DO 162 I=1, KM	00000770
162 R (K, J)=R (K, J) -A (K, I) *R (I, J)	00000780
165 R (K, J)=R (K, J)/DT	00000790
170 CONTINUE	00000800
IF (MOD (IC, 2)) 175, 180, 175	00000810
175 P=-P	00000820
180 D=P	00000830
IF (M) 185, 200, 185	00000840
185 II=N	00000850
DO 190 K=2, N	00000860
KP=II	00000870
II=II-1	00000880
DO 190 J=1, M	00000890
DO 190 I=KP, N	00000900
190 R (II, J)=R (II, J) -A (II, I) *R (I, J)	00000910
200 RETURN	00000920
END	00000930

Subroutine DCUBIC

Description:

Calculates the roots of a cubic equation.

Argument List:

C, R, NRE

C The coefficients of the cubic equation

$$X^3 + C(1)X^2 + C(2)X + C(3) = 0$$

R Roots, if one real root, it is R(1)

NRE The number of real roots

Called by

. DQRTIC

C	SUBROUTINE DCUBIC (C,R,NRE)	00000010
C		00000020
C	SOLVES POLYNOMIAL EQUATION OF THP TYPE	00000030
C	$X^3 + C(1)X^2 + C(2)X + C(3) = 0$	00000040
2		00000050
C	THE COEFFICIENT OF X^3 IS ASSUMED TO BE 1	00000060
C		00000070
C	R CONTAINS THE: ROOTS	00000080
C		09000090
C	NRE CONTAINS THE NUMBER OF REAL ROOTS	00000100
C		00000110
C	IF THERE IS ONE REAL ROOT IT WILL BE	00000120
C	IC P(1), WITH THE COMPLEX ROOTS $R(2) \pm R(3)*I$	00000130
C		00000140
C		00000150
	DIMENSION C(3),R(3)	00000160
	DOUBLE PRECISION C,R,C1SQ,P,Q,DEL,T,A,CRTA,CRTB,PHI3,CON,Y	03000170
	DOUBLE PRECISION B,SQ,HQ	00000180
	$C1SQ = C(1)**2$	00000190
	$P = C(2) - C1SQ/3.D0$	00000200
	$Q = C(3) - (C(2)/3.D0 - 2.D0*C1SQ/27.D0)*C(1)$	00000270
	$DEL = 4.D0*P**3 + 27.D0*Q*Q$	00000220
	$T = C(1)/3.D0$	00000230
5	IF (DEL) 20,10,10	00000240
10	$SQ = DSQRT(DEL/108.D0)$	00000250
	$HQ = .5D0*Q$	00000200
	$A = -HQ + SQ$	00000270
	$B = -HQ - SQ$	00000280
	$CRTA = DSIGN(CABS(A)**.3333333333333333D0,A)$	00000290
	$CRTB = DSIGN(DABS(B)**.3333333333333333D0,B)$	00000300
15	$Y = CRTA + CRTB$	00000310
	$R(1) = Y - T$	00000320
	$R(2) = -.5D0*Y - T$	00000330
	$R(3) = .86602540378443865D0*(CRTA - CRTB)$	00000340
	NRE=1	00000350
	GO TO 40	00000360
20	$PHI3 = DATAN2(DSQRT(-DEL/27.D0),-Q)/3.D0$	09000370
22	$CON = 2.D0*DSQRT(-P/3.D0)$	00000380
30	$R(1) = CON*DCOS(PHI3) - T$	00000390
	$R(2) = -CON*DCOS(1.0471975511965977D0 - PHI3) - T$	00000400
	$R(3) = -CON*DCOS(1.0471975511965977D0 + PHI3) - T$	00000410
	NRE=3	00000420
40	RETURN	00000430
	END	00000440

subroutine DQRTIC

Description:

Calculates the roots of a quartic equation.

Argument List:

C, R, NRE

C The coefficients of the quartic equation
$$x^4 + C(1)x^3 + C(2)x^2 + C(3)x + C(4) = 0$$

R Roots, if 2 real, they are R(1) and R(2)

NRE The number of real roots

Called by:

. SHADOW

Calls Subroutines :

DCUBIC

Subroutine EVALMP (EVALMPSS)

Description :

Evaluates M and $\frac{\partial M}{\partial \underline{z}}$ (see Eqs. (3.23) and (3.36)). The form of M which is coded in EVALMP is that shown in Ref. 5, Table 4. It was also from this form that $\frac{\partial M}{\partial \underline{z}}$ was calculated, while holding F constant when taking partials with respect to a, h, k, p, q .

Argument List:

X, THETA, AMU, AM, PAM, IMFLAG

X Five equinoctial orbital elements

THETA Eccentric longitude

AMU Gravitational constant, μ

AM The matrix $M = \frac{\partial \underline{z}}{\partial \underline{x}}$ (Eq. (3.24))

PAM The partial of M with respect to the orbital elements

IMFLAG Flag, if = 1, AM and PAM are calculated; if = 2, only M is calculated; if = 3, only PAM is calculated

Common Areas :

ORBIT2/X1, Y1, RA, PZ20, PZ26, PZ29, PZ35

Called by:

FCT

C	EVALMP/EVALMPSS	00000010
C		00000020
C		00000030
C	THIS SUBROUTINE EVALUATES THE 5X3 MATRIX M AND THE	00000040
C	5X3X5 PARTIAL OF M WPT X	00000050
C		00000060
C	IF IMFLAG=1, BOTH M (AM) AND ITS PARTIAL (PAM) ARE EVALUATED	00000070
C	IF IMFLAG=2, ONLY M (AM) IS EVALUATED	00000080
C	IF IMFLAG=3, ONLY THO, PARTIAL OF M (PAM) IS EVALUATED	00000090
C		00000100
C		00000110
	SDBBODTINE EVALMP(X, THETA, AMU, AM, PAM, IMFLAG)	00000120
	IMPLICIT REAL*8(A-H,O-Z), INTEGER (I-H)	00000130
	DIMENSION X(5), AM(5,3), PAM(5,3,5)	00000140
	COMMON /ORBIT2/ X1,Y1,RA,PZ20,PZ26,PZ29,PZ35,X1DOT,Y1DOT	00000150
C		00000160
C		00000170
	EN=DSQRT(AMU/X(1)**3)	00000180
	RHO= DSQRT(1.D0- X(2)**2- X(3)**2)	00000190
	BETA= 1.D0/(1.D0 +RHO)	00000200
	CT= DCOS(THETA)	00000210
	ST= DSIN(THETA)	00000220
	RA= 1.D0-X(3)*CP -X(2)*ST	00000230
	ZETA= X(3)*ST-X(2)*CT	00000240
	BETA3= BETA**3/(1.D0 -BETA)	00000250
	X1= X(1)*((1.D0 -X(2)**2*BETA)*CT +X(2)*X(3)*BETA*ST -X(3))	00000260
	Y1= X(1)*((1.D0 -X(3)**2*BETA)*ST +X(2)*X(3)*BETA*CT -X(2))	00000270
	X11=X1	00000280
	Y11=Y1	00000290
	X1DOT= -((1.D0 -X(2)**2*BETA)*ST -X(2)*X(3)*BETA*CT)*EN*X(1)/RA	00000300
	Y1DOT= ((1.D0 -X(3)**2*BETA)*CT -X(2)*X(3)*BETA*ST)*EN*X(1)/RA	00000310
	PZ1= X(1)*[ZETA*(BETA+X(2)**2*BETA3) -X(2)*BETA -ST)*CT/RA]	00000320
	PZ2= -X(1)*[-ZETA*X(2)*X(3)*BETA3 +1.D0 +(ST -X(2)*BETA)*ST/RA]	00000330
	PZ3= X(1)*(-ZETA*X(2)*X(3)*BETA3-1.D0 +(X(3)*BETA -CT)*CT/RA]	00000340
	PZ4= X(1)*(-ZETA*(BETA +X(3)**2*BETA3) +(CT -X(3)*BETA)*ST/RA]	00000350
	IF (IMFLAG.EQ. 3) GO TO 10	00000360
C	IT DO NOT WANT TO EVALUATE PARTIAL OF M, BRANCH TO 10	00000370
	AM(1,1)= 2.D0*X1DOT/(EN**2*X(1))	00000380
	AM(1,2)= 2.D0*Y1DOT/(EN**2*X(1))	00000390
	AM(1,3)=0.D0	00000000
	DUM= RHO/(EN*X(1)**2)	00000410
	AM(2,1)= DUM*(PZ2- X(2)*BETA*X1DOT/EN)	00000420
	AM(2,2)= DUM*(PZ4 -X(2)*BETA*Y1DOT/EN)	00003430
	AM(2,3)= DUM*(X(3)*(X(5)*Y1 -X(4)*X1))/RHO**2	00000440
	AM(3,1)= -DUM*(PZ1 +X(3)*BETA*X1DOT/EN)	00000450
	AM(3,2)= -DUM*(PZ3 +X(3)*BETA*Y1DOT/EN)	00000460
	AM(3,3)= -DUM*(X(2)*(X(5)*Y1 -X(4)*X1)/RHO**2)	00000470
	AM(4,1)=0.D0	00000480
	AM(4,2)=0.D0	00000490
	DUM= (1.D0 +X(4)**2 +X(5)**2)/(2.D0*EN*X(1)**2*RHO)	00000500
	AM(4,3)= DUM*Y1	00000510
	AM(5,1)=0.D0	00000520
	AM(5,2)=0.D0	00000530
	AM(5,3)= DUM*X1	00000540
	IF (IMFLAG.EQ. 2) RETURN	00000550
C	IF WE ONLY WISH TO EVALUATE M THEN PROGRAM RETURNS HERE	00000560
10	CA= DSQRT(AMU/X(1))/RA	00000570
	PZ5= X(2)*BETA3	00000580
	PZ6= X(3)*BETA3	00000590
	PZ9= CA*ST/RA	00000600
	PZ10= CA*CT/RA	00000610

```

PZ20= X(1)*(-2.D0*X(2)*BETA*CT +X(3)*BETA*ST +PZ5*ZETA*X(2)) 00000620
PZ26= X(1)*(X(2)*BETA*ST -1.D0 +PZ6*X(2)*ZETA) 00000630
PZ29= X(1)*(X(3)*BETA*CT -1.90 -PZ5*X(3)*ZETA) 00000640
PZ35= X(1)*(-2.D0*X(3)*BETA*ST +X(2)*BETA*CT -PZ6*X(3)*ZETA) 00000650
PZ11= -X1DOT/(2.D0*X(1)) 00000660
PZ12= -Y1DOT/(2.D0*X(1)) 00000670
DUM1= 1.D0 -RA 00000680
PZ13= -CA*(-2.D0*X(2)*BETA*ST -X(3)*BETA*CT -PZ5*X(2)*DUM1)+PZ9 00000690
1 X1DOT/CA 00000700
PZ14= -CA*(-X(2)*BETA*CT -PZ6*X(2)*DUM1) +PZ10*X1DOT/CA 00000710
PZ15= -CA*(X(3)*BETA*ST +PZ5*X(3)*DUM1) +PZ9*Y1DOT/CA 03000720
PZ16= -CA*(2.D0*X(3)*BETA*CT +X(2)*BETA*ST +PZ6*DUM1*X(3)) 00000730
1 +PZ10*Y1DOT/CA 00000740
DUM= BETA +X(2)*PZ5 00000750
PZ17= 1.D0+ PZ5*X(2)*(3.D0/BETA +1.D0/(1.D0-BETA)) 00000760
PZ18= (2.D0 +PZ17)*PZ5 03000770
PZ19= PZ17*PZ6 00000780
DUM2= X(2)*BETA -ST 00000790
PZ21= -X(1)*(CT*DUM -ZETA*PZ18 +CT*DUM/RA +CT*ST*DUM2/RA**2) 00000800
PZ22= X(1)*(ST*DUM +ZETA*PZ19 -CT*X(2)*PZ6/RA-CT**2*DUM2/RA**2) 00000810
PZ23= BETA3*(3.D0/BETA +1.D0/(1.D0 -BETA)) 00000820
PZ24= PZ23*PZ5 00000830
PZ25= PZ23*PZ6 00000840
PZ27= X(1)*(-CT*X(2)*X(3)*BETA3 +ZETA*X(3)*(BETA3 +X(2)*PZ24) 00000850
1 +(ST*(BETA +X(2)*PZ5))/RA +ST**2*DUM2/RA**2) 00000860
PZ28= X(1)*(ST*X(2)*X(3)*BETA3 +ZETA*X(2)*(BETA3 +X(3)*PZ25) 00000870
1 +X(2)*ST*PZ6/RA +ST*CT*DUM2/RA**2) 00000880
DUM2= X(3)*BETA-CT 00000890
PZ30= X(1)*(CT*X(2)*X(3)*BETA3 -ZETA*X(3)*(BETA3 +X(2)*PZ24) 00000900
1 +CT*X(3)*PZ5/RA +CT*ST*DUM2/RA**2) 00000910
PZ31= X(1)*(-ST*X(2)*X(3)*BETA3 -ZETA*X(2)*(BETA3 +X(3)*PZ25) 00000920
1 +CT*(BETA +X(3)*PZ6)/RA +CT**2*DUM2/RA**2) 00000930
DUM= BETA +X(3)*PZ6 00000940
PZ32= 1.D0 +PZ6*X(3)*(3.D0/BETA +1.D0/(1.D0 -BETA)) 00000950
PZ33= PZ32*PZ5 00000960
PZ34= PZ32*PZ6 +2.D0*X(3)*BETA3 00000970
PZ36= X(1)*(CT*DUM -ZETA*PZ33 -ST*X(3)*PZ5/RA -ST**2*DUM2/RA**2) 00000980
PZ37= X(1)*(-ST*DUM -ZETA*PZ34 -ST*(BETA +X(3)*PZ6)/RA -ST*CT 00000990
1 *DUM2/RA**2) 00001000
DO 20 J=1,2 00001010
20 PAM(1,J,1)= 3.D0*AM(1,J)/(2.D0*X(1)) 00001020
DUM =2.D0*X(1)**2/AMU 00001030
PAM(1,1,2)= PZ13*DUM 00001040
PAM(1,1,3)= PZ14*DUM 00001050
PAM(1,2,2)= PZ15*DUM 00001060
PAM(1,2,3)= PZ16*DUM 00001070
DUM= DSQRT(AMU*X(1)) 00001080
CB=RHO/DUM 00001090
PZ38= -X(2)*CB/RHO**2 00001100
PZ39= -X(3)*CB/RHO**2 00001110
PAM(2,1,1)= AM(2,1)/(2.D0*X(1)) 00001120
PAM(2,1,2)= -CB*BETA*X1DOT/EN +PZ38*AM(2,1)/CB +CB*(PZ27 00001130
1 -X(2)*BETA*PZ13/EN -X(2)*X1DOT*PZ5/EN) 00001140
PAM(2,1,3)= PZ39*AM(2,1)/CB +CB*(PZ28 -PZ6*X(2)*X1DOT/EN 00001150
1 -X(2)*BETA*PZ14/EN) 00001160
PAM(2,2,1)= AM(2,2)/(2.D0*X(1)) 00001170
PAM(2,2,2)= PZ38*AM(2,2)/CB +CB*(PZ36 -BETA*Y1DOT/EN -X(2) 00001180
1 *Y1DOT*PZ5/EN -X(2)*BETA*PZ15/EN) 00001190
PAM(2,2,3)= PZ39*AM(2,2)/CB +CB*(PZ37 -X(2)*Y1DOT*PZ6/EN 00001200
1 -X(2)*BETA*PZ16/EN) 00001210
PAM(2,3,1)= AM(2,3)/(2.D0*X(1)) 00001220

```

	DUM1= X (5) *Y1 -X (4) *X1	00001230
	PAM (2,3,2)= X (3) * (X (5) *PZ29 -X (4) *PZ20) / (RHO* DUM) +X (2) *X (3)	00001240
	1 *DUM1/ (RHO**3*DUM)	00001250
	PAM (2,3,3)= DUM1/ (RHO*DUM) +X (3) * (X (5) *PZ35 -X (4) *PZ26) / (RHO	00001260
	1 *DUM) +X (3) **2*DUM1/ (RHO**3*DUM)	00001270
	PAM (2,3,4)= -X (3) *X1/ (RHO*DUM)	00001280
	PAM (2,3,5)= X (3) *Y1/ (RHO*DUM)	00001290
	PAM (3,1,1)= AM (3,1) / (2.D0*X (1))	00001300
	PAM (3,1,2)= PZ38*AM (3,1) /CB -CB* (PZ21 +X (3) *X1DOT*PZ5/EN	00001310
	1 +X (3) *BETA*PZ13/EN)	00001320
	PAM (3,1,3)= PZ39*AM (3,1) /CB -CB* (PZ22 + (BETA*X1DOT +X (3)	00001330
	1 *X1DOT*PZ6 +X (3) *BETA*PZ14) /EN)	00001340
	PAM (3,2,1)= AM (3,2) / (2.D0*X (1))	00001350
	PAM (3,2,2)= PZ38*AM (3,2) /CB -CB* (PZ30 +X (3) * (Y1DOT*PZ5	00001360
	1 +BETA*PZ15) /EN)	00001370
	PAM (3,2,3)= PZ39*AM (3,2) /CB -CB* (PZ31 + (BETA*Y1DOT +X (3)	00001380
	1 *Y1DOT*PZ6 +X (3) *BETA*PZ16) /EN)	00001390
	PAM (3,3,1)= AM (3,3) / (2.D0*X (1))	00001400
	PAM (3,3,2)= -DUM1/ (RHO*DUM) -X (2) * (X (5) *PZ29 -X (4) *PZ20) /	00001410
	1 (RHO*DUM) -X (2) **2*DUM1/ (RHO**3*DUM)	00001420
	PAM (3,3,3)= -X (2) * (X (5) *PZ35 -X (4) *PZ26) / (RHO*DUM) -X (2) *X (3)	00001430
	1 *DUM1/ (RHO**3*DUM)	00001440
	PAM (3,3,4)= X (2) *X1/ (RHO*DUM)	00001450
	PAM (3,3,5)= -X (2) *Y1/ (RHO*DUM)	00001460
	Z5= (1.D0 +X (5) **2 +X (4) **2) / (2.D0*DUM*RHO)	00001470
	PZ40= -Z5/ (2.D0*X (1))	00001480
	PZ41= X (2) *Z5/RHO**2	00001490
	PZ42= X (3) *Z5/RHO**2	00001500
	PZ43= X (4) / (DUM*RHO)	00001510
	PZ44= X (5) / (DUM*RHO)	00001520
	PAM (4,3,1)= AM (4,3) / (2.D0*X (1))	00001530
	PAM (4,3,2)= PZ41*Y1+Z5*PZ29	00001540
	PAM (4,3,3)= PZ42*Y1 +Z5*PZ35	00001550
	PAM (4,3,4)= PZ43*Y1	00001560
	PAM (4,3,5)= PZ44*Y1	00001570
	PAM (5,3,1)= AM (5,3) / (2.D0*X (1))	00001580
	PAM (5,3,2)= PZ41*X1 +Z5*PZ20	00001590
	PAM (5,3,3)= PZ42*X1 +Z5*PZ26	00001600
	PAM (5,3,4)= PZ43*X1	00001610
	PAM (5,3,5)= PZ44*X1	00001620
	DO 30 K=1,5	00001630
	PAM (1,3,K)=0.D0	00001640
	DO 30 I=4,5	00001650
	DO 30 J=1,2	00001660
30	PAM (I,J,K)=0.D0	00001670
	DO 40 I=1,3	00001680
	DO 40 J=1,2	00001690
	DO 40 K=4,5	00001700
40	PAM (I,J,K)=0.D0	00001710
	RETURN	00001720
	END	00001730

ORIGINAL PAGE IS
OF POOR QUALITY

Subroutine FCT (FCTSS)

Description:

Evaluates what is essentially the preaveraged derivative (see Section 3.3, Reference 1, Eq. (3.33), (3.34)) of the state and costate due to thrusting. Specifically,

$$\left(\frac{\partial H}{\partial \lambda} \frac{dt}{dF} \right) = \dot{\underline{z}} \frac{dt}{dF} \frac{R^2}{a}$$

$$\left[\frac{\partial H}{\partial \lambda} \frac{dt}{dF} + H \frac{\partial}{\partial \lambda} \left(\frac{dt}{dF} \right) \right] \frac{R}{a}$$

The factor $\frac{a}{R^2}$ and the minus sign of Eq. (3.34) and the factor $\frac{1}{2\pi}$ are taken into account in FUNCT. When FCT is called by OUTP, the thrust direction is calculated and then returns prior to the derivative calculation.

Argument List:

F1, F2, z, H, G

F1 Eccentric longitude

F2 Another eccentric longitude

z Five orbital elements and their adjoints

H Preaveraged derivative for $\dot{\underline{z}}$ and $\dot{\underline{\lambda}}$ corresponding to F1

G Preaveraged $\dot{\underline{z}}$ and $\dot{\underline{\lambda}}$ at F_2

Common Areas:

A/A, AMU, PI

ORBIT1/VEC(3), PA

ORBIT2/X1, Y1, RA, PR(2,2), X1DOT, Y1DOT

CCOM/CD(4)

SOL/RS(4)

CON/CB, SB, THETA, CTHETA, STHETA, C2T, C4T, S2T, S4T

Subroutine PCT (FCTSS) (Continued)

Called by:

QUAD, FUNCT, OUTP

Calls Subroutines:

EVALMP, DRTNI

FCT/FCTSS	00000010
C THIS SUBPROGRAM IS CALLED BY THE QUADRATURE PROGRAM AND	00000020
C EVALUATES THE INTEGRAND	00000030
C	00000040
C	00000050
C	00000060
C	00000070
SUBROUTINE FCT (P1,P2,Z,H,G)	00000080
C	00000090
IMPLICIT REAL*8 (A-H,O-Z) , INTEGER (I-N)	00000100
C	09000110
COMMON /A/A,AMU,PI	00000120
COMMON /ORBIT1/ YEC (3) ,PA	00000130
COMMON /ORBIT2/X1,Y1,RA,PR (2,2) ,X1DOT,Y1DOT	00000140
COMMON /CCOM/CD (4)	00000150
COMMON /SOL/RS (4)	00000160
COMMON /CON/CB,SB,THETA,CTHETA,STHETA,C2T,C4T,S2T,S4T	00000170
C	00000180
EXTERNAL CONE	00000190
DIMENSION Z (10) ,G (10) ,H (10) ,AM (5,3) ,PAM (5,3,5) ,X (5) ,PRA (5) ,	00000200
1 PPA (5) ,DRSP (3) ,DRSQ (3) ,PVEC (3)	00000210
C	00000220
N=0	00000230
F=F1	00000240
C	00000250
C EVALUATE M AND PARTIAL OF M YRT STATE	00000260
C	00000270
DO 5 I=1,5	00000280
5 X (I)=Z (I)	00000290
10 CALL EVALHP (X,F,AMU,AM,PAM,1)	00000300
-	00000310
EVALUATE THE COMMON SCALAR FACTOR	00000320
C	00000330
CT=DCOS (F)	00000340
ST=DSIN [P]	00000350
C EVALUATE THE PARTIAL OF BA WRT X	00900360
PRA (1)= 0.D0	00000370
PRA (2)= -ST	00000380
PBA (3)= -CT	00000390
PRA (4)= 0.D0	00000400
PRA (5)= 0.D0	00000410
C	00000420
C EVALUATE M TRANSPRESS LAUBDA (PRIMER VECTOR)	00000430
C	00000440
DO 20 I=1,3	00000450
PVEC (I)= 0.D0	00000460
DO 20 J= 1,5	00000470
20 PVEC (I)= PVEC (I)*AM (J,I) *Z (J+5)	00000480
C *****	00000490
C WRITE (6,1000) .F,Z,CT,ST,RA, PRA ,PYEC	00000500
C *****	00000510
C	00000520
19 ABVEC= 0.D0	00000530
DO 22 I=1,3	00000540
22 ABVEC= ABVEC+PVEC (I)*PVEC (I)	00000550
ABVEC= DSQRT (ABVEC)	00000560
DO 23 I=1,3	00000570
23 PVEC (I)= PVEC (I)/ABVEC	00000580
C *****	00000590
C WRITE (6,1000) PYEC,ABVEC	00000600
*****	00000610

*

ORIGINAL PAGE
OF POOR QUALITY

C		00000620
C		00000630
C	PRIMER CONE ANGLE.	00000640
	CB=0.D0	00000650
	DO 53 I= 1,3	00000660
50	CB= PVEC (I)*RS (I)+CB	00000670
	SB= 1.D-10	00000680
	IF (DEBS (CB) .LT. 1.D0) SB=DSQRT (1.D0-CB*CB)	00000690
C		00000700
C	PARTIAL OF RS WBT P,Q	00000710
	D= 2.D0/(1.D0+Z (4) *Z (4) +Z (5) *Z (5))	00000720
	DRSP (1)= (-RS (2) *Z (5) -RS (3)) *D	00000730
	DRSQ (1)= RS (2) *Z (4) *D	00000740
	DRSP (2)= RS (1) *Z (5) *D	00000750
	DRSQ (2)= (-RS (1) *Z (4) +RS (3)) *D	00000760
	DRSP (3)= RS (1) *D	00000770
	DRSQ (3)= -RS (2) *D	00000780
C		00000790
C	THRUST CONE ANGLE	00000800
	THG= PI/2.D0	00000810
	IF [DABS (SB) .GT. 1.D-8) THG= DATAN2 ((3.D0*CB+DSQRT (8.D0*SB*SB	00000820
1	+9.D0*CB*CB) ,-4.D0*SB)	00000830
	IF (THG.LT.CD (4)) THG= (1.00001D0*CD (4))	00000840
	CALL DRTNI (THETA,PT,DPT,CONE,THG,1.E-10,100,IER)	00000850
	IF (IER.GT.0) URITE (6,1001) IER,THG,THETA,PT,DPT	00000860
	IF (THETA.GT.PI) URITE (6,1002) THG,THETA	00000870
	IF (THETA.LT.CD (4)) THETA= CD (4)	00000880
	STHETA= DSIN (THETA)	00000890
	CTHETA= DCOS (THETA)	00000900
	C2T= DCOS (2.D0*THETA)	00000910
	C4T= DCOS (4.D0*THETA)	00000920
C		00000930
C		00000940
	PCO= STHETA/SB	00000950
	RCO= CTHETA-PCO*CB	00000960
	DO 70 I=1,3	00000970
70	VEC (I)= RCO*RS (I)+PCO*PVEC (I)	00000980
	ABVEC= 0.D0	00000990
	DO 72 I=1,3	00001000
72	ABVEC= ABVEC+VEC (I)*VEC (I)	00001010
	ABVEC= DSQRT (ABVEC)	00001020
	DO 74 I=1,3	00001030
74	VEC (I)= VEC (I)/ABVEC	00001040
C		00001050
C	ACCELERATION FACTOR AND PARTIAL	00001060
	PA= CD (1) +CD (2) *C2T+CD (3) *C4T	00001070
	DUMMY= 4.D0*CTHETA*(CD (2) +4.D0*CD (3) *C2T)	00001080
	DO 80 I= 1,3	00001090
80	PPA (I)= 0.D0	00001100
	UPRSP= 0.D0	00001110
	UPRSQ= 0.D0	00001120
	DO 85 I=1,3	00001130
	UPRSP= UPRSP+VEC (I)*DRSP (I)	00001140
85	UPRSQ= UPRSQ+VEC (I) *DRSQ (I)	00001150
	PPA (4)= DUMMY*UPRSP	00001160
	PPA (5)= DUMMY*UPRSQ	00001170
C		00001180
C	*****	00001190
C	URITE (6,1000) X1,Y1,RS,CB,SB,DRSP,DRSQ,THETA,CTHETA,	00001200
C	1 STHETA,C2T,C4T,S2T,S4T,PA,PPA,THG,PT,DPT	00001210
C	*****	00001220

C		00001230
C	EVALUATE FUNCTION	00001240
90	IF (F2.EQ.0.D0) RETURN	00001250
	DO 100 I=1,5	00001260
	G(I)= 0.D0	00001270
	DO 100 J=1,3	00001280
100	G(I)= G(I)+AM(I,J)*VEC(J)	00001290
	HZ = 0.D0	00001300
	DO 120 I=1,5	00001310
120	HZ = HZ +Z(I+5)*G(I)	00001320
	DUM1= HZ *RA	00001330
	DO 130 I= 1,5	00001340
	G(I+5)= 0.D0	00001350
	DO 130 J=1,3	00001360
	DO 130 L=1,5	00001370
130	**G(I+5)= G(I+5)-Z(L+5)*PAM(L,J,I)*VEC(J)	00001380
C	** *****	00001390
C	BRITE (6,1000) G, DUM1,PRA,PPA	00001400
C	*****	00001410
	DO 140 I=1,5	00001420
140	G(I+5)= PA*(G(I+5)*RA-HZ*PRA(I))-PPA(I)*DUM1	00001430
C	*****	00001440
C	WRITE (6,1000) G,HZ	00001450
C	*****	00001460
C		00001470
	DUMMY= RA*PA	00001480
	DO 150 I= 1,5	00001490
150	G(I)= G(I)*DUMMY	00001500
C		00001510
	I? (M.EQ.1) RETURN	00001520
	DO 200 I= 1,10	00001530
200	H(I)= G(I)	00001540
	F=F2	00001550
	M=1	00001560
	GO TO 10	00001570
C1000	FORMAT (1H0,1P10D12.4)	00001580
1001	FORMAT (1H0, 'ERROR IN NERTON ITERATOR FOR CONE ANGLE'/' IER = ',	00001590
1	I2, 'THG= ',1PD20.12, 'THETA= ',1PD20.12, 'FT= ',1PD20.12,	00001600
2	'DFT= ',1PD20.12)	00001610
1002	FORMAT (1H0, 'ERROR--THETA GREATER THAN 180 DEG'/' THG= ',	00001620
1	1PD20.12, 'THETA= ',1PD20.12)	00001630
	END	00001640

Subroutine FUNCT (FUNCTSS)

Description:

Called principally by the differential equation integrator to evaluate the averaged derivative for the full state and costate. Can take into account sun location, shadowing, and oblateness,

Argument List:

X, Z, DERZ

X Time

Z State and costate

DERZ Derivative of state and costate

Common Areas:

A/A, AMU, PI

J2/AJ2

SHAD/FEN, FEX, DFEN(5), DFEN(5), ISHAD

BCOM/B(9)

IS/ISUN, ISON

SOL/RSUN(3), RS

NU/XNU

Q/IQ

Called by:

RK4, TRAJ

Calls Subroutines:

SUN, SHADOW, QUAD, FCT, OBLATE

ORIGINAL PAGE 15
OF POOR QUALITY

C FUNCT/FUNCTSS	00000010
	00003020
i	00000030
C SOLAR SAIL	00000040
C THIS SUBROUTINE IS AN INTERFACE BETWEEN THE INTEGRATOR ROUTINE	00000050
C AND THE QUADRATURE ROUTINE.	00000060
C INCLUDES SHADOW EFFECT.	00000070
2 THIS ROUTINE ADDS THE EFFECT OF OBLATENESS (J2) TO THE DERIV.	00000080
C OBLATE CALCULATES THE EFFECT OF J2. RETURNED AS DZJ2.	00090099
C Z IS A VECTOR OF THE AVERAGED STATE AND COSTATZ	03000100
C DERZ IS THE AVERAGED DERIVATIVE OF Z	00000110
C INCLUDES PENALTY FUNCTION.	00000120
C	00000130
C	00000140
C	00000150
SUBROUTINE FUNCT(X, Z, DERZ)	00000160
C	00000170
C	00000180
IMPLICIT REAL*8 (A-H, O-Z)	03000190
COMMON /A/A, AMU, PI	09000200
COMMON /J2/J2	00000210
CCHMON /SHAD/ PEN, FEX, PFEN(5), DFEX(5), ISHAD	00000220
COMMON /BCOM/B(9)	00000230
CCHMON /IS/ISUN, ISON	00000240
COMMON /SOL/RSUN(3), RS	00000250
CCHMON /Q/IQ	00000260
COMMON /NU/XNU	00000270
C	00000280
C	00000290
DIMENSION Z(10), DERZ(10), G(10), H(10), DZJ2(10), GEX(10), GEN(10)	00000300
DIMENSION DZ10(10)	00000310
	00000320
EXTERNAL PCT	00000330
C	00000340
C SET UP COEFFS OF COSF AND SINFI IN X1 AND Y1 AND PARTIALS	00000350
C	00000360
C	00000370
BETA= 1.D0/(1.D0+DSQRT(1.D0-Z(2)**2-Z(3)**2))	00000380
B(1)= 1.D0-Z(2)**2*BETA	00000390
E(2)= Z(2)*Z(3)*BETA	00000400
B(3)= 1.D0-Z(3)**2*BETA	00000410
BETA3= BETA**3/(1.D0-BETA)	00000420
A1= Z(2)**2*BETA3	00000430
A2= Z(3)**2*BETA3	00000440
A3= BETA+A1	00000450
A4= BETA+A2	00000460
B(4)= -Z(2)*(BETA+A3)	00000470
B(5)= Z(3)*A3	00000480
B(6)= -Z(2)*A2	00000490
E(7)= -Z(3)*A1	00000500
B(8)= Z(2)*A4	00000510
B(9)= -Z(3)*(BETA+A4)	00000520
C	00000530
C	00000540
C	00000550
QEX= 0.D0	00000560
QEN= PI+PI	00000570
C	00000580
CALL SUN(X, Z)	00000590
C	00000600
	00000610

23	FAC= A/(PI+PI)	00000620
	IF (ISON.EQ.1) FAC=FAC/RS**2	00000630
C		00000640
C		00000650
	ISHAD= 0	00000660
	IF (ISON.EQ.0) GO TO 24	00000670
C		00000690
C	SHADOW INFLUENCE	00000690
C		00000700
	CALL SHADOW (Z)	00000710
	IF (ISHAD.EQ.0) GO TO 24	00000720
	QEN= ?EN	00000730
	QEX= HEX	00000740
	IF (QEN.LT.QEX) QEN= QEN+2.DO+PI	00000750
C		00000760
24	AIF= (QEN-QEX)/DFLOAT(IQ)	00000770
	IPV=0	00000780
	F1= QEX	00000790
	DO 26 I=1,10	00000800
26	DERZ(I)= 0.DO	00000810
27	F2= F1+AIF	00000820
	CALL QUAD (F1,F2,FCT,DZ10,Z,G,H,10)	00000830
	IPV= IPV+1	00000840
	F1= F2	00000850
	DO 37 I=1,10	00000860
37	DERZ(I)= DERZ(I)+DZ10(I)	00000870
	IF (IPV.LT.IQ) GO TO 27	00000880
C		00000990
31	DO 32 I=1,10	00000900
32	DERZ(I)= DERZ(I)*FAC	00000910
C	PENALTY FUNCTION EFFECT	00000920
	IF (XNU.EQ.0.DO) GO TO 38	00000930
	E= DSQRT(Z(2)*Z(2)+Z(3)*Z(3))	00000940
	DUM=1.DO-E	00000950
	EL2=XNU/(Z(1)*Z(1)*DUM*DUM)	00000960
	DERZ(6)= -EL2/Z(1)+DERZ(6)	00000970
	DUM= EL2/(1.DO-E)	00000980
	DUM1=1.DO	00000990
	IF (E.GT.1.D-10) DUM1=Z(2)/E	00001000
	DERZ(7)= DERZ(7)+DUM1*DUM	00001010
	IF (E.GT.1.D-10) DUM1=Z(3)/E	00001020
	DERZ(8)= DERZ(8)+DUM1*DUM	00001030
C		00001040
C		00001050
C		00001060
38	IF (ISHAD.EQ.0) GO TO 90	00001070
C		00001090
C	SHADOW INPLUEHCZ	00001090
C		00001100
	CALL FCT (QEN,QEX,Z,GEN,GEX)	00001110
	HWX=0.DO	00001120
	HWN= 0.DO	00001130
	DO 40 I=1,5	00001140
	HWX= HWX+Z(I+5)*GEX(I)	00001150
40	HWN= HWN+Z(I+5)*GEN(I)	00001160
	HWX= HWX*FAC	00001170
	HUN= HWN*FAC	00001180
	DO 50 I=1,5	00001190
50	DERZ(I+5)= DERZ(I+5)-HWN*DFEN(I)+HWX*DFEX(I)	00001200
C		00001210
C		00001220

```

90  IF (AJ2.LE.0.D0) RETURN
C  OBLATENESS BPPECT
C
    CALL OBLATE(AJ2,Z,DZJ2,1)
    DO 120 I=1,10
120  DERZ(I)= DERZ(I)+DZJ2(I)
    RETURN
C
    END

```

```

00001230
00001240
00001250
03001260
00001270
00001280
00001290
00001300
00001310
00001320

```


Subroutine INPUT (INPUTSS)

Description:

Reads initial data, **sets** initial values, and prints initial information.

Common Areas:

XMMM/ZLO(5), STEP(6), ZERF(6)
ELEM/ZPO(5), ZPF(5)
INT/IPR, IDIM, IDIM2, NIMAX
TRA/IFMAX, DT, UEB, EW(10)
UNITS/UTS, UTM, UTH, UTD, UTKM, DTR, UTKG, UTKW, UTMS2
T/TE, TO
A/A, AMU, PI
WF/WF(5)
J2/AJ2
TC/NOP
JD/TL
IS/ISUN, ISON
F/FLIM, RSTEP
ORBIT/NORB
CONSTR/ICON
PLOT/IPL
TERRA/AE, EC, UU, ENS, AN, ECLMTX(3,3), EQUMTX(3,3)
CCOM/CD(4)
Q/IQ

Called by:

CONTL

Calls Subroutines:

PLANET

ORIGINAL PAGE 15
OF POOR QUALITY

C INPUT/INPUTSS		00000010
		03000020
		00000030
C SOLAR SAIL--ORBIT RAISING AND ESCAPE		00000040
C THIS SUBPROGRAM IS CALLED BY CONT1 AND READS AND PRINTS		00000050
C ALL INITIAL DATA AS WELL AS SETS INITIAL CONSTANTS.		00000060
C THE UNITS ARE BASED ON INTERNAL MU=1.0, INTERNAL DISTANCE		00000070
C UNIT=1 PLANET RADII. a CIRCULAR		00000080
C ORBIT AT 1 PLANET RADII WOULD HAVE A PERIOD OF 2 PI INTERNAL		00000090
C TIME UNITS,		00000100
C INPUT		03000110
C IPNUM - PLANET NUMBER 1-MERCURY, 2-VENUS, 3-EARTH, 4-MARS		00000120
C INITIAL ORBIT		00000130
C A (KM)		00000140
C E		00000150
C I (DEG)		00000160
C LONG. OF NODE (DEG)		00000170
C ARG. OF PERICENTER (DEG)		00000180
C INITIAL GUESSES		00000190
C LAMBDA A		00000200
C LAMBDA H		00000210
C LAMBDA K		00000220
C LAMBDA P		00000230
C LAMBDA Q		00000240
C FINAL CRBIT		00000250
C A		00000260
C E (NOT USED IF NOP=3)		00000270
C I (NOT USED IF NOP=3)		00000280
C NODE (NOT USED IF NOP=2 OR 3)		60000290
C PERICENTER (NOT USED IF NOP=2 OR 3)		00000300
C		00000310
C TF2 (DAYS), GUBSS FOR FINAL TIRE		00000320
C AC (MM/S**2) CHARACTERISTIC ACCELERATION		00000330
C TL JULIAN DATE AT INITIAL TIME		00000340
C IRDFLG	NOMINAL	00000350
C 1 END OF INPUT		00000360
C 2 IPR, PRINT FLAG	0	00000370
C 3 NIMAX, MAX. NO. OF ITERATIONS	20	00000380
C 4 TFMAX2 (DAYS), MAX. TF	500.	00000390
C 5 DT2 (DAYS), TIME STEP FOR D.E.	10.	00000400
C 6 DEB, UPPER ERROR BOUND FOR D.E.	1.D10	00000410
C 7 EW, ERROR WEIGHTS FOR D.E.	1.,1,1,1,1,0,...	00000420
C 9 GM (KM**3/SEC**2) GRAV.CONST.	SET IN PLANETS	00000440
C 10 NOP = 1, 5 O.E. SPECIFIED AT TF	1	00000450
C = 2, 3 O.E. SPECIFIED AT TF		00000460
C = 3, SEMIMAJOR AXIS SPECIFIED AT TP		00000470
C 11 SETS OBLATENESS, AJ2, = 0.D0	SET IN PLANETS	00000480
C 12 STEP(8), STEP SIZE FOR NUMERICAL DIFF.	1.D-6	00000490
C RSTEP = 0, STEP AS FRACTION IN ITER	0	00000500
C = 1, STEP AS CONSTAHT IN ITER		00000510
C 13 ISON = 0, SHADOW EFFECT @FF	0	00000520
C = 1, SHADOW EFFECT ON		00000530
C 14 ISUN = 0 SUN DIST EFFECT ON THRUST OFF	0	000005110
C = 1 EFFECT OH		00000550
C 15 CD(1), COEFFICIENT IN ACCELERATION FACTOR	.5	00000560
C CD(2)	.5	00000570
C CD(3)	.0	00000580
C 16 IE, IF = 1, EQUATORIAL FRAME, SOLAR MOTION	1	00000590
C = 2, EQUATORIAL FRAME, NO SOLAR MOTION		00000600
C = 3, ECLIPTIC FRAME, SCLAR ROTION		00000610
C = 4, ECLIPTIC FRAKE, NO SOLAR MOTION		00000620

	= 5, PLANETARY FRAME, SOLAR MOTION		00000630
	= 6, PLANETARY FRAME, NO SOLAR MOTION		00000640
C 17	FLIM, NORM LIMIT IN ITERATION ROUTINE	1.D-06	00000550
C 18	XNU, PERICENTER PENALTY FUNCTION FACTOR	0.	00000660
C 19	NORB = 0, NO ORBIT PRINT	0	00000670
	= 1, ..., 999, ORBIT PRINT ON NORB POINTS OF ORBIT		00000690
C 20	SETS IPL=1 SO THAT PLOT DATA IS STORED	0	00000690
C 21	IQ, NUMBER OF QUADRATURES (1-10)	2	00000700
C			00000710
C			00000720
	SUBROUTINE INPUT		00000730
C			00000740
	IMPLICIT REAL*8(A-H,O-Z) , INTEGER (I-N)		00000750
C			00000760
	NAMelist /UN/UTKM,UTS,UTD,UTKG,UTKW,UTMS2		00000770
C			00000780
	COMMON /XMMH/ZLO(5), STEP(6) , ZERP(6)		00000790
	COMMON /ELEM/ZPO(5), ZPF(5)		00000800
	COMMON /INT/IPR,IDIM,IDIM2,NIMAX		00000810
	COMMON /TRA/TFMAX,DT,UEB,EW(10)		00000820
	COMMON /UNITS/UTS,UTM,UTH,UTD,UTKM,DTR,UTKG,UTKW,UTMS2		00000830
	COMMON /T/TF,T0		00000810
	COMMON /A/A,AMU,PI		00000850
	COMMON /WF/WF(5)		00000860
	COMMON /J2/AJ2		00000870
	COMMON /TC/NOP		00000880
	COMMON /JD/TL		00000890
	COMMON /IS/ISUN,ISON		00000900
	COMMON /P/PLIM,KSTEP		00000910
	COMMON /ORBIT/ NORB		00000920
	COMMON /CCOM/CD(4)		00000930
	COMMON /Q/IQ		00000940
	COMMON /PLOT/IPL		00000950
	COMMON /TERRA/AE,EC,UU,ENE,AN,ECLMTX(3,3),EQUMTX(3,3)		00000960
	COMMON /PLNTS/ IPNUM		00000970
	COMMON /NU/XNU		00000980
C			00000990
	DIMENSION W(5),PCHAR(4)		00001000
C			00001010
	DATA PCHAR/'MERCURY ','VENUS ','EARTH ','MARS '/'		00001020
C			00001030
C	INTEGER CONSTANTS		00001040
C			00001050
	IDIM=10		00001060
	IDIM2=5		00001070
	IDIM3=6		00001080
	IPR=0		00001090
	ITF=3		00001100
	NIMAX=20		00001110
	NOP= 1		00001120
	ISON= 0		00001130
	ISUN= 0		00001140
	KSTEP= 1		00001150
	NORB= 0		00001160
	IQ= 2		00001170
	IE= 1		00001180
	IPL= 0		00001190
C			00001200
C	REAL CONSTANTS		00001210
C			00001220
	AMU=1.0D0		00001230

	UEB= 1.0D+10	00001240
	DO 10 I=1, IDIM2	00001250
	EW(I+IDIM2)= 0.D0	00001260
10	EW(I)= 1.0D0	00001270
	DO 12 I=1, IDIM3	00001280
12	STEP(I)= 1.D-2	00001290
	STEP(IDIM3)= 1.D-6	00001300
	DT2= 10.D0	00001310
	DTR= .017453292519943296D0	00001320
	PI= 3.1415926535897932D0	00007330
	TFMAX2= 500.0D0	00001340
	TFMIN= 0.0D0	00001350
	T02=0.0D0	00001360
	TL= 0.D0	00001370
	FLIM= 1.0-06	00001380
	CD(1)= .5D0	00001390
	CD(2)= .5D0	00001400
	CD(3)= 0.D0	00001410
	XNU= 0.D0	00001420
C		00001430
C	ALL READ STATEMENTS FOLLOW	00001440
C		00001450
	REAC (5,1006) IPNUM	00001460
	URITE (6,2065) PCHAR(IPNUM)	00001470
	IF (IPNUM.LT.1.OR.IPNUM.GT.4) GO TO 240	00007480
	READ (5,1001) (W(I), I=1, IDIM2)	00001490
	WRITE (6,3001) (W(I), I=1, IDIM2)	00001500
	BEAD (5,1001) (ZL0(I), I=1, IDIM2)	00001510
	WRITE (6,3001) (ZL0(I), I=1, IDIM2)	00001510
	READ (5,1001) (WF(I), I=1, 5)	00001530
	WRITE (6,3001) (WF(I), I=1, 5)	00001540
	READ (5,1001) TF2	00001550
	WRITE (6,3001) TP2	00001560
	READ (5,7001) AC	00001570
	WRITE (6,3001) AC	00001580
	READ (5,1001) TL	00001590
	WRITE (6,3001) TL	00001600
C	TL MUST BE BETWEEN ABOUT A.D. 1950 AND 2000	00001610
	IF ((TL.LT.2.433D6).OR.(TL.GT.2.452D6)) GO TO 230	00001620
C	CALL PLANETS TO SET UP DEFAULTS	00001630
	CALL PLANET(GM)	00001640
C		00001650
20	BEAD (5,1002) IRDFLG	09001660
	WRITE (6,3002) IRDFLG	00001670
	IP ((IRDFLG.GT.21).OR.(IRDFLG.IT.1)) GO TO 200	00001680
25	GO TO (150, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 1 49, 50, 51), IRDFLG	00001690
		00001700
C		00001710
C	THESE VALUES ARE READ ONLY IF INDICBTCL BY IRDFLG	00001720
C		00001730
32	READ (5,1002) IPB	00001740
	WRITE (6,3002) IPR	00001750
	IF (IPR.LT.0) GO TO 210	00001760
	GO TO 20	00001770
33	BEAD (5,1002) NIMAX	00001780
	WRITE (6,3002) NIMAX	00001790
	IF ((NIMAX.LT.0).OR.(NIMAX.GT.50)) GO TO 220	00001800
	GO TO 20	00001810
34	READ (5,1001) TFMAX2	00001820
	WRITE (6,3001) TFMAX2	00007833
	IF ((TFMAX2.LT.0.D0).OR.(TFMAX2.GT.1.D3)) GO TO 220	00001840

	G3 TO 20	00001850
35	READ (5,1001) DT2	00001860
	WRITE (6,3001) DT2	00001870
	IF ((DT2.LE.0.D0).OR.(DT2.GT.1-03)) GO TO 220	00001880
	GC TO 20	00001890
36	READ (5,1001) UEB	00001900
	WRITE (6,3001) UEB	00001910
	GC TO 20	00001920
37	READ (5,1003) EW	00001930
	WRITE (6,3003) EW	00001940
	GO TO 20	00001950
38	READ (5,1001) UTKM	00001960
	WRITE (6,3001) UTKM	00001970
	GO TO 20	00001980
39	READ (5,1001) GM	00001990
	WRITE (6,3001) GM	00002000
	GO TO 20	00002010
40	READ (5,1002) NOP	03002020
	WRITE (5,3002) NOP	00002030
	IF ((NOP.LT.1).OR.(NOP.GT.3)) GO TO 220	00002040
	GC TO 20	00002050
41	AJ2= 0.D0	00002060
	GO TO 20	00002070
42	READ (5,1001) (STEP(I),I=1,IDIM3)	00002090
	WRITE (6,3001) (STEP(I),I=1,IDIM3)	00002090
	READ (5,1002) KSTEP	00002100
	WRITE (6,3002) KSTEP	00002110
	IF ((KSTEP.LT.0).OR.(KSTEP.GT.1)) GO TO 220	00002120
	GO TO 20	00002133
43	READ (5,1002) ISON	00002140
	WRITE (6,3002) ISON	00002150
	IF ((ISON.LT.0).OR.(ISON.GT.1)) GO TO 220	00002160
	GO TO 20	00002170
44	READ (5,1002) ISUN	00002180
	WRITE (6,3002) ISUN	00002190
	IF ((ISUN.LT.0).OR.(ISUN.GT.1)) GO TO 220	00002200
	GO TO 20	00002210
45	READ (5,1001) (CD(I),I=1,3)	00002220
	WRITE (6,3001) (CD(I),I=1,3)	00002230
	IP (DABS(CD(1)+CD(2)+CD(3)-1.D0).GT.1.D-8) GO TO 220	00002240
	GO TO 20	00002250
46	READ (5,1002) IE	00002260
	WRITE (6,3002) IE	03002270
	IF ((IS-LT-1).OR.(IE.GT.6)) GO TO 220	00002280
	IF (IE.EQ.2 .OR. IE.EQ.4 .OR. IE.EQ.6) ENE=0.D0	00002290
	IP (IE.LE.2) GO TO 20	00002300
	EQUmtx(1,1)=1.D0	00002310
	EQUmtx(2,1)=0.D0	00002320
	EQUmtx(3,1)=0.D0	00002330
	EQUmtx(1,2)=0.D0	00002340
	EQUmtx(2,2)=1.D0	00002350
	EQUmtx(3,2)=0.D0	00002360
	EQUmtx(1,3)=0.D0	00002370
	EQUmtx(2,3)=0.D0	00002380
	EQUmtx(3,3)=1.D0	00002390
	IP (IE.LE.4) GO TO 20	00002400
	UU=0.D0	00002410
	ECLmtx(1,1)=1.D0	00002420
	ECLmtx(1,2)=0.D0	00002430
	ECLmtx(1,3)=0.D0	00002440
	ECLmtx(2,1)=C.D0	00002450

ECLMTX(2,2)=1.D0	09002460
ECLMTX(2,3)=0.D0	00002470
ECLMTX(3,1)=0.D0	09002420
ECLMTX(3,2)=0.D0	03302490
ECLMTX(3,3)=1.D0	00002500
GO TO 20	00002510
47 READ (5,1003) FLIM	03002520
WRITE (6,3003) FLIK	00002530
IF (FLIM.LT.0.D0) GO TO 220	00002540
GO TO 20	00002550
48 READ (5,1001) XNU	00002560
WRITE (6,3001) XNU	00002570
GO TO 20	00002580
49 READ (5,1005) NORB	00002590
WRITE (6,3005) NORB	00002600
IF ((NORB.LT.0).OR.(NORB.GT.999)) GO TO 220	00002610
GO TO 20	00002620
50 IPL=1	00002630
GO TO 20	00002640
51 READ (5,7002) IQ	00002650
WRITE (6,3002) IQ	00002660
IF ((IQ.LT.1).OR.(IQ.GT.10)) GO TO 220	00002670
GO TO 20	00002680
C	00002690
C	00002700
C	00002710
C	00002720
C TIME VALUES ARE CHANGED FROM CAPS TO OTHER UNITS	00002730
C	00002740
150 UTS= DSQRT(UTKM**3/GM)	00002750
UTM=UTS/60.D0	00002760
UTH=UTS/3600.D0	00002770
UTD=UTH/24.D0	03002780
C	00002790
C NOW MULTIPLY ENE BY UTD, SINCE WE KNOW WHAT IT IS, THIS	00002800
C USED TO BE DONE IN EARTH/PLANET	00002810
ENE=ENE*UTD	00002820
C	00002830
TO= T02/UTD	00002840
TP= TF2/UTD	00002850
TFMAX= TP*UTS	00002860
DT= DT2/UTD	00002870
T01= T0*UTS	00002880
TF1= TF*UTS	00002890
TFMAX1= TFEAX*UTS	00002900
DT1= DT*UTS	00002910
C	00002920
C MORE CONVERSIONS	00002930
C	00002940
UTMS2= UTKM*1.D3/(UTS**2)	00002950
UTKG= 1.D3	00002960
UTKW= UTKG*UTMS2*UTKM/UTS	00002970
C	00002980
C CALCULATE LOWER LIMIT ON THETA	00002990
CD(4)=0.D0	00003000
IF ((CD(2)+CD(3)).EQ.0.D0) GO TO 155	00003010
CK3= 8.D0*CD(3)	00003020
CK1= CD(1)-CD(2)+CD(3)	00003030
CK2= 2.D0*CD(2)-CK3	00003040
IF (CD(3).EQ.0.D0) CD(4)= DARCOS(-DSQRT(CK1/CK2))	00003050
IF (CD(3).NE.0.D0) CD(4)= DARCOS(-DSQRT((-1.D0+DSQRT(1.D0-4.D0*	00003060

1	CK1*CK3/CK2**2))*CK2/(2.D0*CK3)))	00003070
155	CONTINUE	00003080
C		00003090
C		00003100
C	THE PRINTING OF ALL INITIAL VALUES FOLLOWS	00003110
C		00003120
	WRITE (6,2000)	00003130
	WRITE (6,2065) PCHAR(IPNUM)	00003140
	IF (ISON.GE.1) RRITE (6,2042)	00003150
	WRITE (6,2002)	00003160
	WRITE (6,2003)	00003170
C		00003180
158	WRITE (6,2004) (W(I),I=1,5)	00003190
C		00003200
C	CHANGE FROM CLASSICAL O.E. TO EQUINOCTIAL O.E.	00003210
	DO 160 I=3,5	00003220
160	W(I)=W(I)*DTR	00003230
	ZP0(1)=W(1)/UTKM	00003240
	ZP0(2)=W(2)*DSIN(W(5)+W(4))	00003250
	ZP0(3)=W(2)*DCOS(W(5)+W(4))	00003260
	ZP0(4)=DTAN(W(3)/2.0D0)*DSIN(W(4))	00003270
	ZP0(5)=DTAN(W(3)/2.0D0)*DCOS(W(4))	00003280
C		00003290
	A=(AC*1.D-3)/UTMS2	00003300
C		00003310
C		00003320
	WRITE (6,2005)	00003330
	WRITE (6,2004) (ZP0(I),I=1,5)	00003340
C		00003350
	WRITE (6,2038) AC,A	00003360
	A=A/AE**2	00003370
	CD4=(PI-CD(4))/DTR	00003380
	WRITE (6,2063) (CD(I),I=1,3),CD4	00003390
	IF (ISUN.NE.0) WRITE (6,2044)	00003400
C		00003010
C	WRITE FINAL CONDITIONS, CHANGE TO EQUINOCTIAL FINAL COND.	00003420
C		00003430
	WRITE (6,2006)	00003440
	ZPF(1)=WF(1)/UTKM	00003450
	GO TO (165,170,180),NOP	00003460
C		00003470
165	WRITE (6,2003)	00003480
	WRITE (6,2004) (WF(I),I=1,5)	00003490
	DO 165 I=3,5	00003500
166	WF(I)=WF(I)*DTR	00003510
	ZPF(2)=WF(2)*DSIN(WP(5)+WF(4))	00003520
	ZPF(3)=WF(2)*DCOS(WP(5)+WF(4))	00003530
	ZPF(4)=DTAN(WP(3)/2.0D0)*DSIN(WP(4))	00003540
	ZPF(5)=DTAN(WP(3)/2.0D0)*DCOS(WP(4))	00003550
	DC 167 I=3,5	00003560
167	UP(I)=WF(I)/DTR	00003570
	WRITE (6,2005)	00003580
	WRITE (6,2004) (ZPF(I),I=1,5)	00003590
	GO T3 190	00003600
C		00003610
17C	ZPF(2)=WF(2)	00003520
	ZPF(3)=DABS(DTAN(WP(3)*DTR/2.0D0))	00003630
	ZPF(4)=0.D0	00003640
	ZPF(5)=0.D0	00003650
	WRITE (6,2031)	00003660
	WRITE (6,2004) (WF(I),I=1,3)	00003670

	WRITE (6,2032)	000036'33
	WRITE (6,2004) (ZPF(I), I=1, 3)	00003690
	GO TO 190	00003700
180	C3= -GM/WF(1)	00003710
	WRITE (6,2064) WF(1), ZPF(1), C3	00003720
C		00003730
C		00003740
190	WRITE (6,2007)	00003750
196	WGITZ (6,2011) ZLO	00003760
	WRITE (6,2008)	00003770
	FRITE (6,2009) TF2, TF1, TF	00003780
	WRITE (6,2037) TL	00003790
	WRITE (6,2036) BJ2	00003800
	WRITE (6,2013)	00003810
	FRITE (6,2009) T02, T01, T0	00003820
	WRITE (6,2015)	00003830
	WRITE (6,2009) TFMAX2, TFMAX1, TFMAX	00003840
	WRITE (6,2010) KSTEP	00003850
	WRITE (6,2011) STEP	00003860
	PRITE (6,201 E)	00003870
	WRITE (6,2009) DT2, DT1, DT	00003880
	WRITE (6,2017) DEB	00003890
	WRITE (6,2018)	00003900
	WRITE (6,2019) EW	00003910
	WRITE (6,2058) IQ	00003920
	WRITE (6,2020) IDIM	00003930
	WRITE (6,2022) NIMAX	00003940
	RRITE (6,2048) PLIK	00003950
	WRITE (6,2026) UTKM	00003960
	WRITE (6,2027) GM	00003970
	WRITE (6,2069) XNU	00003980
	IF (IE.EQ.1) WRITE (6,2059)	00003990
	IF (IE.EQ.2) RRITE (6,2060)	00004000
	IF (IE.EQ.3) WRITE (6,2061)	00004010
	IF (IE.EQ.4) WRITE (6,2062)	00004020
	IF (IE.EQ.5) FRITE (6,2067)	00004030
	IF (IE.EQ.6) WRITE (6,2068)	00004040
	WRITE (6, UN)	00004050
	RETURN	00004060
200	WRITE (6,2023) IRDFLG	00004070
	STOP	00004080
210	WRITE (6,2024) IPR	00004090
	STOP	00004100
220	URITE (6,2025) IRDPLG	00004110
	STOP	00004120
230	#RITE (6,2037) TL	00004130
	STOP	00004140
240	WRITE (6,2066) IPNUM	00004150
	STOP	00004160
C		00004170
1001	FORMAT (F25.15)	00004180
1002	FORHAT (I2)	00004190
1003	FORMAT (506.1)	00004200
1004	FORMAT (P18.11)	00004210
1005	FORHAT (13)	00004220
1006	FORMAT (11)	00004230
2000	FORMAT (1H1, 19X, ' MINIMUM TIME OPTIMAL TRAJECTORY PROGRAM FOR PLA	00004240
	RNETOCENTRIC SOLAR SAIL SPIRAL' / 1H0, 50X, ' ESCAPE OR CAPTURE')	03004250
2001	FORHAT (1H0, 53X, 13H MINIMUM TIME)	00004260
2002	FORMAT (34H0 THE INITIAL ORBITAL ELEHENTS ARE)	00004270
2003	FORMAT (1H0, 10X, 6HA (KM), 19X, 1HE, 20X, 7HI (DEG), 10X, 18HLON ASC NODE	00004280

1 (DEG), 6X, 15HARG PERIG (DEG))	00004290
2004 FORMAT (1P5D23.14)	00004300
2005 FORMAT (1H0, 5X, 14HA (PLANET RAD), 16X, 1HH, 22X, 1HK, 22X, 1HP, 22X, 1HQ)	00004310
2006 FORMAT (40H0 THE DESIRED FINAL ORBITAL ELEMENTS ARE)	00004320
2007 FORMAT (32H0 INITIAL GUESSED PARAJETZBS ARE)	00004330
2008 FORMAT (21H0 FINAL TIME ESTIMATE)	00004340
2009 FORMAT (1H ,10X, 1PD22.15, 7H DAYS =, 1PD22.15, 10H SECONDS =, 1PD22.15	00004350
1, 6H UNITS)	00004360
2010 FORMAT (50H0 STEP SIZE FOR NUMERICAL DIFFERENTIATION, KSTEP =, I2)	00004370
2011 FORMAT (1P5D23.14)	00004380
2013 FORMAT (17H0 INITIAL TIME IS)	00004390
2015 FORMAT (10H0 TMAX IS)	00004400
2016 FORMAT (27H0 TIME STEP FOR INTEGRATION)	00004410
2017 FORMAT (36H0 UPPER ERROR BOUND ON INTEGRATION =, 1PD20.10)	00004420
2018 FORMAT (35H0 ERROR WEIGHTS FOR INTEGRATION ARE)	00004430
2019 FORMAT (1P10D8.1)	00004440
2020 FORMAT (13H0 DIHENSION =, I5)	00004450
2022 FORMAT (32H0 MAXIMUM NUMBER OF ITERATIONS =, I5)	00004460
2023 FORMAT (44H0 XRDFLG SHOULD BE BETWEEN 1 AND 21, IT IS =, I5)	00004470
2024 FORMAT (28H0 IPR SHOULD BE < 0, IT IS =, I5)	00004480
2025 FORMAT (27H0 BAD INPUT DATA, IRDFLG =, I3)	00004490
2026 FORMAT (18H0 1 PLANET RADIUS =, F14.5, 3H KM)	00004500
2027 FORMAT (11H0 MU (GM) =, F16.5, 13H KM**3/SEC**2)	00004510
2031 FORMAT (1H0, 10X, 6HA (KM), 19X, 1HE, 20X, 7HI (DEG))	00004520
2032 FORMAT (1H0, 5X, 14RH (PLANET RAD), 9X, 15HSQRT(H**2+K**2), 8X,	00034530
1 15HSQRT(P**2+Q**2))	00004540
2037 FORMAT (32H0 JULIAN DATE AT INITIAL TIME IS, F20.8)	00004550
2036 FORMAT (6H0 J2 =, 1PD15.7)	00004560
2038 FORMAT ('0 CHARACTERISTIC ACCELERATION = ', F18.11, 'MM/S**2 ='	00004570
1 , 1PD15.7, ' INTERNAL UNITS')	00004580
2042 FORMAT (24H0 SHADOW EFFECT INCLUDED)	00004590
2044 FORMAT ('0 SUN DISTANCE EFFECT INCLUDED')	00004600
2046 FORMAT (28H0 INITIAL A (PLANET RADII) =, 1PD25.14)	00004610
2048 FORMAT (23H0 NORM LIMIT IN ITER =, 1PD12.5)	00004620
2058 FORMAT ('0 IQ =, I3)	00004630
2059 FORMAT ('0 EQUATORIAL FRAME, SOLAR MOTION')	00004640
2060 FORMAT ('0 EQUATORIAL FRAME, NO SOLAR MOTION')	00004650
2061 FORMAT ('0 ECLIPTIC FRAME, SOLAR MOTION')	00004660
2062 FORMAT ('0 ECLIPTIC FRAME, NO SOLAR MOTION')	00004670
2063 FORMAT ('0 ACCELERATION FACTOR COEFFICIENTS = ', 1P3D20.12/	00004680
1 '0 MAXIMUM THETA (DEG) = ', 1PD20.12)	00004690
2064 FORMAT ('0 SEHHAJOR AXIS = ', 1PD23.14, ' KR = ', 1PD23.14,	00004700
1 ' P.R.; C3 = ', 1PD23.14, ' KM**2/SEC**2')	00004710
2065 FORMAT (' SPACE VEHICLE IS IN ORBIT ABOUT ', A8)	00004720
2066 FORMAT (' IPNUH SHOULD BE BETWEEN 1 AND 4, IT IS ', I2)	00004730
2067 FORMAT ('0 PLANETARY ORBITAL FRAME, SOLAR MOTION')	00004740
2068 FORMAT ('0 PLANETARY ORBITAL FRAME, NO SOLAR MOTION')	00004750
2069 FORMAT ('0 PEEXCENTER PEGALTY FUNCTION PXCTOR = ', 1PD10.2)	00004760
3001 FORMAT (1H , F25.15)	00004770
3002 FORMAT (1H , I2)	00004780
3003 FORMAT (1H , 5D6.1)	00004790
3004 FORXST (1H , F18.11)	00004800
3005 FORMAT (1H , I3)	00004810
END	00004820

Subroutine ITER (NRSS, MODNRSS)

Description :

A Newton-Raphson iterator. It calculates a nominal trajectory, receiving the error in the final condition called \underline{y} . It then varies the free initial conditions \underline{x} (including the value of t_f) to get a sensitivity matrix or partial derivative matrix. By inverting this matrix and pre-multiplying \underline{y} , new values of \underline{x} are obtained and a new nominal trajectory run. This is continued until the norm of the errors in the final conditions is less than some inputted tolerance or until the convergence fails. There is provision for halving the Δx 's several times.

An optional Modified Newton-Raphson iterator does not calculate a new partial derivative matrix by running neighboring trajectories at each iteration but rather updates the partial derivative matrix using the old one and the Lx terms. If convergence is poor a new matrix is calculated by running neighboring trajectories, however,

Argument List:

KOUNT, NI, FUNCT, PRTN

KOUNT Number of trajectory calls

NI The number of iterations

FUNCT Dummy subroutine name - that is called to calculate the "y" for the iterator (in our case this is TRAJ)

PRTN Dummy subroutine name - used to print iteration info (in our case this is called PRTN)

Common Areas:

XMM/X(5), XS(6), Y(6)

INT/IPR, IDIM, IDIM2, MAXNOI

T/TE, TO

DY/DYDT(6)

F/FLIM, KSTEP

Called by:

CONIL

Calls Subroutines :

DCROUT, PRTN, TRAJ

C	ITER	00000010
C	NEWTON RAPHSON NRSS	00000020
C		03003030
C	SPECIAL VERSION FOR 6X6 PARTIAL DER. MATRIX, 6 DIM. Y.	00000040
C		00003050
	SUBROUTINE ITER(KCUNT,NI,FUNCT,PRTN)	00000060
C		00000070
	IMPLICIT REAL*8(A-H,O-Z), INTEGER (I-N)	00003080
C		00000090
C		00000100
C	X VALUES OF THE INDEPENDENT VARIABLES (INITIAL,CURRENT,FINAL)	00000110
C	XS STEP SIZE TO PERTURB XS TO COMPUTE PARTIAL DERIVATIVES	00000120
C	Y VALUES OF THE DEPENDENT VARIABLES (CURRENT, FINAL)	00000130
	COMMON/XMMH/X(5),XS(6),Y(6)	00000140
	COMMON /INT/IPR,IDIM,IDIM2,MAXNCI	00000150
	COMMON /T/TF,TO	00000160
	CCHMON /DY/DYDT(6)	00000170
	COMMON /F/PLIM,KSTEP	00000180
C		00000190
		00000200
C	ARRAYS USED INTERNALLY BY THE ITERATOR	00000210
	DIMENSION YNOM(6),XN(5),P(6,6),COEF(6)	00000220
	EQUIVALENCE (YNOM,COEF)	00000230
C		00000240
	N= 5	00000250
	M= 6	00000260
	NI=1	00000270
	KCUST=0	00001280
	CALL FUNCT	00000290
	KCUNT=KCUNT+1	00000300
	F0=0.D0	00000310
	DO 15 I=1,M	00000320
15	F0=F0+Y(I)**2	00000330
10	DO 16 I=1,N	00000340
	XN(I)=X(I)	00000350
16	YNOM(I)=Y(I)	00000360
	YNOM(M)= Y(M)	00000370
	TFN=TF	00000380
	CALL PRTN(KCUNT,NI)	00000390
	WRITE(6,1011)F0	00000400
	IF(F0.LE.FLIM) GO TO 90	00000410
	IF (NI.GT.MAXNCI) GO TO 80	00000420
C	CCOMPUTE NUMERICAL PARTIAL DERIVATIVES	00000430
	WRITE (5,1013)	09000540
	DO 17 I=1,M	00000450
17	P(I,M)= DYDT(I)	00000460
	DO 25 J=1,N	00000470
	TEMP=X(J)	00000480
	STEP=XS(J)*DABS(X(J))	00000490
	IF ((DABS(X(J)).LT.1.D-10).OR.(KSTEP.EQ.1)) STEP=XS(J)	00030500
	IF (DABS(X(J)).LT.1.D-10) WRITE (6,1014)	00000510
	X(J)=X(J)+STEP	03000520
	CALL FUNCT	00000530
	WRITE(6,1000)X(J)	00000540
	WRITE(6,1001) (Y(I),I=1,M)	00000550
	DO 20 I=1,M	00000560
20	P(I,J)=(Y(I)-YNOM(I))/STEP	00000570
25	X(J)=TEMP	00000580
	KCUNT=KCUNT+N	00000590
	WRITE(5,1002)	00000600
	DO 30 I=1,M	00000610

	WRITE(6,1001) (P(I,J),J=1,M)	00000620
30	CONTINUE	00000630
	DO 35 I=1,M	00000640
35	COEF(I)=-YNOM(I)	00000650
	CALL DCROUT(P,COEF,DET,0.00,M,1,IND)	00000660
	IF(IND.NE.0) GO TO 85	00000670
	WRITE(6,1015) DET	00000680
	DO 40 I=i,n	00000690
40	IF (DABS(COEF(I)) .LT. 1.D-12) COEF(I) = 0.00	09000700
	WRITE(6,1003) (COEF(I),I=1,N)	00000710
	SN= COEF(M)	00000720
	WRITE(6,1012) SN	00000730
	DO 50 J=1,N	00000740
50	X(J)=XN(J)+COEF(J)	00000750
	TF=TFN + SN	00000760
	IHALV=0	00000770
51	IF (DABS(SN).GT. (.25D0*TFN)) GO TO 60	00000780
	CALL FUNCT	00000790
	KOUNT=KOUNT+1	00000800
	F1=0.00	00000810
	DO 52 I=1,M	00000820
52	F1=F1+Y(I)+*2	00000830
	WRITE(6,1010) F1	00000840
	IF(F1.LT.F0) GO TO 55	00000850
60	WRITE(6,1006)	00000860
	IF (IHALV.EQ.10) GO TO 95	00000870
	IHALV=IHALV+1	00000880
	DO 53 J=1,N	00000890
	COEF(J)=COEF(J)/2.00	00000900
	WRITE(6,1000) COEF(J)	00000910
53	X(J)=XN(J)+COEF(3)	00000920
	SN= SN/2.00	00000930
	WRITE(6,1012) SN	00000940
	TF= TFE + SN	00000950
	GO TO 51	00000960
55	IF(NI-MAXNOI) 70,70,80	00000970
70	NI=NI+1	00000980
	F0=F1	00000990
	GO TO 10	00010000
80	NI=9999	00010010
	WRITE(6,1006)	00010020
	RETURN	00010030
85	NI=9999	00010040
	WRITE(6,1007)	00010050
	RETURN	00010060
90	WRITE(6,1005) PO	00010070
	RETURN	00010080
95	NI=9999	00010090
	WRITE(6,1009)	00011000
	RETURN	00011010
1000	FORMAT(/1X,1PD23.15)	00011020
1001	FORMAT(1X,1PD23.15)	00011030
1002	PORKAT(21HOPARTIAL DERIV MATRIX)	00007140
1003	FORMAT(11HODELX:S ARE/(1X,1PD23.15))	00001150
1005	FORMAT(4HOF0=,1PD22.15,23HCASE CONVERGED...FERTIG)	00001160
1006	FORMAT(38H0EXCEEDED MAXIMUM NUMBER OF ITERATIONS)	00001170
1007	FORMAT(16HOMATRIX SINGULAR)	00001180
1008	FORMAT(11HODELX:S ARE)	00001190
1009	FORMAT(19HOMETHOD CANNOT WORK)	00001200
1010	FORMAT(4HOF1=,1PD23.15)	00005210
1011	FORMAT(4HOF0=,1PD23.15)	00001220

1012 FORMAT (10H0 DEL TF =,1PD23.15)
1013 FORMAT (40H0 X(I)*DX(I) FOLLOWED BY CORRESPONDING Y)
1014 FORMAT (24H0 X(I)=0, SO DX(I)=XS(I))
1015 FORHAT (15H0 DETERMINENT =,1PD23.15)
END

00001230
00001240
00001250
00001260
00001270

C	ITER	00000070
C	MCDNR/MCDNRSS MODIFIED N-R ITERATOR	00000020
C		00000030
C		00000040
C	6X6 VERSION	00000050
C		00000060
C	SUBROUTINE ITER(KOUNT,NI,FUNCT,PRTN)	00000070
C		00000080
C	IMPLICIT REAL*8(A-H,O-Z)	00000090
C		00009100
C		00000110
C		00000120
C	X VALUES OF THE INDEPENDENT VARIABLES (INITIAL,CURRENT,FINAL)	00000130
C	XS STEP SIZE TO PERTURB X:S TO COMPUTE PARTIAL DERIVATIVES	00000140
C	Y VALUES OF THE DEPENDENT VARIABLES (CURRENT,FINAL)	00000150
	COMMON/XMMM/X(5),XS(6),Y(6)	00000160
	COMMON /INT/IPR,IDIM,IDIM2,MAXNOI	00000170
	COMMON /I/TF,T0	03000180
	COMMON /DY/ DYDT(6)	00000190
	CCHHON /P/FLIM,KSTEP	00000200
C		00000210
	DIBEPSION YNOK(6),XN(5),P(6,6),COEF(6),DYDTN(6)	00000220
	N=5	03000230
	M=6	00000240
	18=1	00000250
	ICONS=1	00000260
	ISW=0	00000270
	NI=1	00000280
	KOUNT=0	00000290
	CALL PUNCT	00000300
	KOUNT=KOUNT+1	00000310
	F0=0.D0	03000320
	DO 15 I=1,M	00000330
15	F0=F0+Y(I)**2	00000340
9	DO 16 I=1,N	00000350
	DYDTN(I)=DYDT(I)	00000360
	XN(I)=X(I)	00000370
16	YNOM(I)=Y(I)	00000380
	YNOM(M)=Y(M)	00000390
	TFN=TF	00000400
	DYDTN(M)=DYDT(M)	00000410
10	CALL PRTN(KOUNT,NI)	00000420
	WRITE(6,1011)F0	00000430
	IF(F0.LE.FLIM)GO TO 90	00000440
	IF(NI.GT.MAXNOI)GO TO 80	00000450
	IF(ISW.NE.0)GO TO 27	00000460
C	CCHPUTE NUMERICAL PARTIAL DERIVATIVES	00000470
	DO 17 I=1,M	00000480
17	P(I,M)=DYDT(I)	00000490
	WRITE(6,1013)	00000500
	DO 25 J=1,N	00000510
	TEMP=X(J)	00000520
	STEP=XS(J)*DABS(X(J))	00000530
	IF((DABS(X(J)).LT.1.D-10).OR.(KSTEP.EQ.1))STEP=XS(J)	00000540
C	IF(DABS(X(J)).LT.1.D-10)WRITE(6,1014)	00000550
	X(J)=X(J)+STEP	00000560
	CALL FUNCT	00003570
	WRITE(6,1000)X(J)	00000580
	WRITE(6,1001)(Y(I),I=1,M)	00000590
	DO 20 I=1,M	00000600
20	P(I,J)=(Y(I)-YNOM(I))/STEP	00000610

25	X(J)=TEMP	00000620
	KCOUNT=KOUNT+N	00000630
27	WRITE (6,1002)	00000640
	DO 30 I=1,M	00000650
	WRITE (6,1001) (P(I,J),J=1,M)	00000660
30	CONTINUE	00000670
	DO 35 I=1,M	00000680
35	COST(I)=-YNOM(I)	00000690
	CALL DCRUT(P,COEP,DET,0.D0,M,1,IND)	00000700
	IF(IND.NE.0)GO TO 85	00000710
	WRITE (6,1015) DET	00000720
	DO 40 I= 1,M	00000730
40	IF (DABS (COEP (I)).LT.1.D-10) COEP(I)= 0.D0	00000740
C		00000750
	IF (DABS(XN(1)).GT.1.D2) GO TO 47	00000760
	RATS= 1.D0	00000770
	DO 45 I= 1,5	00000780
	RAT= CABS (COEP(I))/(.8D0*DABS(XN(I))+.1D0)	00000790
	IF (RAT.GT.RATS) RATS= RAT	00000800
45	CONTINUE	00000810
	DO 46 I= 1,8	00000820
46	COEP(I)= COEP(I)/RATS	00000830
C	WRITE (5,1016) RATS	00000840
C		00000850
47	WRITE: (6,1003) (COEP(I),I=1,N)	00000860
	SH= COEP(M)	00000870
	WRITE (6,1012) SI	00000880
	DO 50 J=1,N	00000890
50	X(J)=XN(J)+COEP(J)	00000900
	TF=TFN+SN	00000910
	IHALV=0	00000920
51	CALL FUHCT	00000930
	KOUNT=KOUNT+1	00000940
	F1=0.D0	00000950
	DO 52 I=1,M	00000960
52	F1=F1+Y(I)**2	00000970
	WRITE (6,1010) P1	00000980
	IF(F1.LT.F0)GO TO 55	00000990
	WRITE (6,1008)	00001000
	IF(IHALV.EQ.6)GO TO 95	00001010
	IHALV=IHALV+1	00001020
	DO 53 J=1,N	00001030
	COEP(J)=COEP(J)/2.D0	00001040
	WRITE (6,1000) COEP(J)	00001050
53	X(3)=XH(J)+COEP(J)	00001060
	SN=SN/2.D0	00001070
	WRITE (6,1012) SN	00001080
	TF=TFN+SN	00001090
	GO TO 51	00001100
55	IF(NI-MAXNOI) 70,70,80	00001110
70	NI=NI+1	00001120
	ICONS=NI	00001130
	P0=P1	00001140
	SUMDX=0.D0	00001150
	DO 76 J=1,M	00001160
76	SUMDX=COEP(J)**2+SUMDX	00001170
	DO 77 I=1,M	00001180
	DO 77 J=1,M	00001190
	P(I,J)=P(I,J)+(Y(I)*COEP(J))/SUMDX	00001200
77	CONTINUE	00001210
	ISW=1	00001220

GO TO 9	00001230
80 NI=9999	00001240
WRITE(6,1006)	00001250
RETURN	00001260
85 NI=9999	00001270
WRITE(6,1007)	00001280
RETURN	00001290
90 WRITS(6,1005) F0	00001300
RETURN	00001310
95 IF(NI.EQ.1.OR.I8.EQ.10.OR.ICONS.NE.NI) GO TO 100	03001320
ICONS=ICONS+1	00001330
I8=I8+1	00001330
DO 96 J= 1,N	00001350
DYDT(J)= DYDTN(J)	00001360
X(J)= XN(J)	00001370
96 Y(J)= YNOM(J)	00001380
Y(M)= YNOM(E)	00001390
DYDT(M)=DY DTN(M)	00001400
TF= TFN	00001410
ISW=0	00001420
URITE(6,1004)	00001430
GO TO 10	00001440
100 NI=9999	00001450
BRITE(6,1009)	00001460
RETURN	00001470
1000 FORMAT(/1X,1PD23.15)	00001480
1001 FORMAT(1X,1P5D23.15)	00001490
1002 FORMAT(21H0PARTIAL DEBIY KATRIX)	00001500
1003 FORMAT(11H0DELX:S ARE/(1X,1PD23.15))	00001510
1004 FORMAT(35H0FORM NEW PARTIAL DERIVATIVE MATRIX)	00001520
1005 FORMAT(4H0F0=,1PD22.15,23HCASE CONVERGED...FERTIG)	00001530
1006 FORMAT(38H0EXCEEDED MAXIMUM NUMBER OF ITERATIONS)	00001540
1007 FORMAT(16H0MATRIX SINGULAR)	00001550
1008 FORMAT(11H0DELX:S ARE)	00001560
1009 FORHAT(19H0METHOD CANNOT YORK)	00001570
1010 FORMAT(4H0F1=,1PD23.15)	00001580
1011 FORMAT(4H0F0=,1PD23.15)	00001590
1012 FORMAT(10H0 DEL TF =,1PD23.15)	00001600
1013 FORHAT(40H0 X(I)+DX(I) FOLLOWED BY CORRESPONDING Y)	00001610
1014 FORMAT(24H0 X(I)=0, SO DX(I)=XS(I))	00001620
1015 FORKAT(15H0 DETERMINENT =,1PD23.15)	00001630
1016 FORWBT(8H0 RAPS =,1PD23.15)	00001640
END	00001650

Subroutine OBLATE

Description:

Calculates the oblateness effect on $\dot{\underline{z}}$ and $\underline{\lambda}$ (see Section 3.5, Reference 1).

Argument List:

AJ2, Z, DZJ2, IJ —

AJ2 The J_2 coefficient

Z Orbital elements and adjoints

DZJ2 The oblateness contribution to the state and
costate derivative

IJ Flag, always = 1

Called by :

FUNCT

C OBLATE	00000010
C	00000020
C THIS SUBPROGRAM EVALUATES THE AVERAGED DERIVATIVE OF THE	00000030
C STATE AND COSTATE DUE TO THE OBLATENESS, J2 TERM	00000040
C ASSUMES EARTH EQUATORIAL RADIUS=1, MU=1. IF NOT C1 MUST BE	00000050
C MODIFIED.	00000060
C IF IJ=2, EVALUATE DZ(I), I=1,5 ONLY.	00000070
C	00000080
C	00000090
C	00000100
SUBROUTINE OBLATE(AJ2,Z,DZJ2,IJ)	00000110
C	00000120
IMPLICIT REAL*8(A-H,O-Z)	00000130
DIMENSION Z(10),DZJ2(10),PJ(4,5)	03000140
C	00000150
C	00000160
C	00000170
C1= 1.5D0*AJ2/Z(1)**3.5	00000180
B1=Z(4)**2+Z(5)**2	00000190
B7= 1.D0-6.D0*B1 + 3.D0*B1**2	00000200
D2= 1.D0 - Z(2)**2 -Z(3)**2	00000210
B2= 1.D0/D2**2	00000220
B6= 1.D0/(1.D0+E1)	00000230
C2= B2*B6	00000240
B4= 1.D0 - B1	00000250
C3= C1*C2	00000260
D16= B6*B7	00000270
D3= C3*D16	00000280
C	00000290
DZJ2(1)= 0.D0	00000300
DZJ2(2)= Z(3)*D3	00000310
DZJ2(3)= -Z(2)*D3	00000320
C	00000330
D4=B4*C3	00000340
C	00000350
DZJ2(4)= -Z(5)*D4	00000360
DZJ2(5)= Z(4)*D4	00000370
C	00000380
IF (IJ.EQ.2) RETURN	00000390
C	00000400
D5= -3.5D0*C1/Z(1)	00000410
B8= C2*D16*D5	00000420
C	00000430
PJ(1,1)=Z(3)*B8	00000440
PJ(2,1)= -Z(2)*B8	00000450
C	00000460
B12= C2*B4*D5	00000470
C	00000480
PJ(3,1)= -Z(5)*B12	00000490
PJ(4,1)= Z(4)*B12	00000500
C	00000510
D2= .25D0*D2**3	00000520
D7= B6*Z(2)/D2	00000530
B9= C1*D16	00000540
C	00000550
PJ(1,2)= Z(3)*B9*D7	00000560
PJ(2,2)= -B9*(Z(2)*D7 + C2)	00000570
C	00000580
B13= C1*D7*B4	03000590
C	00000600
PJ(3,2)= -Z(5)*B13	00000610

	PJ(4,2) = Z(4)*B13	00000620
C		00000630
	D9= B6*Z(3)/D2	00000640
C		00000650
	PJ(1,3) = B9*(Z(3)*D9 + C2)	00000660
	PJ(2,3) = -Z(2)*B9*D9	00000670
C		00000680
	B14= C1*B4*D9	00000690
C		00000700
	PJ(3,3) = -Z(5)*B14	00000710
	PJ(4,3) = Z(4)*B14	00000720
		00000730
	D10= -2.D0*B2*B6**2	00000740
	D11=Z(4)*D10	00000750
	D12= -12.D0*B4	00000760
	D13= C1*B6	00000770
	D15=C2*D12	00000780
	B10= D13*(2.D0*B7*D11 + Z(4)*D15)	00000790
C		00000800
	PJ(1,4) = Z(3)*B10	00000810
	PJ(2,4) = -Z(2)*B10	00000820
C		00000830
	B15= B4*D11 - 2.D0*Z(4)*C2	00000840
C		00000850
	PJ(3,4) = -Z(5)*C1*B15	00000860
C		00000870
	B16= C2*B4	00000880
C		00000890
	PJ(4,4) = C1*(B16 + Z(4)*B15)	00000900
C		00000910
	D14= Z(5)*D10	00000920
	D15= C2*D12	00000930
	B11= D13*(2.D0*B7*D14 + Z(5)*D15)	00000940
C		00000950
	PJ(1,5) = Z(3)*B11	00000960
	PJ(2,5) = -Z(2)*B11	00000970
C		00000980
	B17= D14*B4 - 2.D0*Z(5)*C2	00000990
C		00001000
	PJ(3,5) = -C1*(B16 + Z(5)*B17)	00001010
	PJ(4,5) = Z(4)*C1*B17	00001020
C		00001030
C		00001040
	DO 10 J=1,5	00001050
	DZJ2(J+5) = 0.D0	00001060
	DO 10 I=1,4	00001070
10	DZJ2(J+5) = DZJ2(J+5) -Z(I+6)*PJ(I,J)	00001080
	RETURN	00001090
	END	00001100

Subroutine **OUTP** (OUTPSS)

Description:

Used to print information at each time step of the differential equation integration of the trajectory. Optionally it can calculate and print information for several points on an individual orbit such as thrust direction, **sun** angles.

Argument List:

T, Z, DERZ, IHLF, IDIM, PRMT

T Time

Z State and costate vector

DERZ State and costate derivative

IHLF Flag indicating error in integrator

IDIM Dimensions of state plus costate

PRMT Parameter vector containing initial and final time,
 time step, the upper error bound, and a fifth element
 not used

Common Areas:

UNITS/UTS, UTM, UTH, UTD, UTKM, DTR, UTKG, UTKW, UTMS2

INT/IPR, ID, IDIMZ, NIMAX

A/A, AMU, PI

SHAD/FEN, FEX, DFEN(5), DFEX(5), ISHAD

IS/ISUN, ISON

SOL/RS(3), RSUN

ORBIT/NORB

ORBIT1/U0(3), PA

ORBIT2/X1, Y1, RA, PR(2,2), XIDOT, YIDOT

CCOM/CD(4)

CON/CB, SB, THETA, DUB(6)

PLOT/IPL

ORIGINAL PAGE 16
OF POOR QUALITY

Subroutine OUPP (OUTPSS) (Continued)

Called by :

RK4

Calls Subroutines:

SUN, SHADOW, FCT, VCP

C	OUTP/OUTPSS	09000010
C		00000020
C	SCLAB SAIL	00000030
C	THIS IS TRE OUTP PROGRAM FOR THE R-K	00000040
C	INTEGRATOR	00000050
C	EQUINOCTIAL OE. AND COSTATE ARE USED.	00000060
C	INCLUDES SHADOW TIME.	00000070
C		00000080
C		00000090
	SUBROUTINE OUTP (T,Z,DERZ,IHLF,IDIM,PRMT)	00000700
C		09000110
	IMPLICIT REAL*8 (A-H,O-Z) - INTEGER (I-N)	00000120
	COMMON /UNITS/UTS,UTM,UTH,UTD,UTKM,DTR,UTKG,UTKW,UTMS2	00000130
	COMMON /INT/IPR,ID,IDIM2,NIMAX	00000140
	COMMON /A/A,AKU,PI	00000150
	COMMON /SHAD/ FEN,PEX,DFEN(5),DPEX(5),ISHAD	00000160
	COMMON /IS/ISON,ISON	00000770
	COMMON /SOL/RS(3),RSUN	00000180
	COMMON /ORBIT/NORB	00000190
	COMMON /ORBIT1/UO(3),PA	00000200
	COMMON /ORBIT2/X1,Y1,RA,PR(2,2),X1DOT,Y1DOT	00000210
	COMMON /CCOM /CD(4)	00000220
	COMMON /CON/CE,SE,THETA,DUB(6)	00000230
	COMMON /PLOT/IPL	00000240
C		00000250
	DIMENSION PRMT(5),Z(10),DERZ(10),W(5)	00000260
	DIMENSION E1(3),E2(3),H1(10),G1(10),VEL(3)	00000270
C		00000280
	INTEGER FRSTSW	00000290
	DATA PRSTSW /0/	00000300
C		00000310
	IF (IPR.EQ.0) EETURY	00000320
	IF (T.EQ.PRMT(1)) N=0	00000330
	IF (T.EQ.PRMT(1)) M=0	00000340
	M=M+1	00000350
	IF ((T.LT. (.9999999999D0*DFLOAT(M) * (FRET(2)-PRMT(1)) /DFLOAT(IPR)))	00000360
1	AND. (IHLF.LT. 11).AND. (T.LT. (.9999999999D0*PRMT(2)))) RETURN	00000370
		00000380
	CALL SUN(T,Z)	00000390
	ACR= (A*UTMS2)*1.D3	00000400
	IF (ISON.EQ.0) GO TO 5	00000410
5	CALL SHADQY(Z)	00000420
	IF (ISON.EQ.1) ACR= ACR/RSUN**2	00000430
C		00000440
	TS=UTS*T	00000450
	TM=UTM*T	00000460
	TH=UTH*T	00000470
	TD=UTD*T	00000480
	H= 0.D0	00000490
	DO 7 I=1,IDIM2	00000500
7	H= H + Z (I+IDIM2) *DERZ (I)	00000510
	W(1)= Z(1)*UTKM	00000520
	W(2)=0.D0	00000530
	DUMMY=Z(2)**2+Z(3)**2	00000540
	IF (DUMMY.GT. 1.D-40) W(2)=DSQRT(DUMMY)	00000550
	W(3)=0.D0	00000560
	DUMMY=Z(4)**2 + Z(5)**2	00000570
	IF (DUMMY.GT. 1.D-40) W(3)= 2.D0*DATAN(DSQRT(DUMMY)) /DTR	00000580
	W(4)=0.D0	00000590
	IF ((Z(4).NE.0.D0).OR. (Z(5).NE.0.D0)) W(4)=DATAN2(Z(4),Z(5)) /DTR	00000600
	W(5)= 0.D0	00000610

	IF ((Z(2).NE.0.D0).OR.(Z(3).NE.0.D0)) W(5)=DATAN2(Z(2),Z(3))/DTR	00000620
	W(5)=W(5)-W(4)	00000630
	IDIM3=IDIM2+1	00000640
C		00000650
C		00000660
C		00000670
C		00000680
	WRITE (6,1001)	00000690
	YWRITE (6,1002)	00000700
	WRITE (6,1003) T, TS, TH, TD, N	00000710
	WRITE (6,1004)	00000720
	URITE (6,1005) (Z(I),I=1,IDIM2)	00000730
	YWRITE (6,1014)	00000740
	URITE (6,1005) W	00000750
	WRITE (6,1006)	00000760
	WRITE (6,1005) (Z(I),I=IDIM3,IDIM)	00000770
	URITE (6,1007)	00000780
	WRITE (6,1005) (DERZ(I),I=1,IDIM2)	00000790
	URITE (6,1008)	00000800
	WRITE (6,1005) (DERZ(I),I=IDIM3,IDIM)	00000810
	WRITE (6,1009)	00000820
	PER=2.D0*PI*DSQRT(Z(1)**3/AMU)*UTH	00000830
	BP= W(1)*(1.D0+W(2))	00000840
	PE= W(1)*(1.D0-W(2))	00000850
	C3= (-AMU/Z(1))^(UTKM/UTS)**2	00000860
	WRITE (6,1010) R,PER,PE,AP,ACR,C3	00000870
C	IF PLOT DATA DESIRED, PUT IT OUT	00000880
	IF (IPL.NE.0.AND. FRSTSW.EQ.0) WRITE (8) HOB	00000890
	FRSTSW=1	00000900
	I? (IPL.NE.0) YWRITE (8) TD,Z,W,H,PER,PE,AP,ACR,C3	00000910
C		00000920
	IP (ISHAD.EQ.0) GO TO 30	00000930
	IF (PEX.LT.FEN) PEX=PEX+2.D0*PI	00000940
	TSHAD= (PEX-PEW+Z(2)*(DCOS(PEX)-DCOS(PEW))-Z(3)*(DSIN(PEX)	00000950
	1 -DSIN(PEW)))*PER/(2.D0*PI)	00000960
	FPER= TSHAD/PER	00000970
	TSHAD= TSHAD*60.D0	00000980
	WRITE (6,1013) TSHAD, FPER	00000990
	PEXD= PEX/DTR	00001000
	FEND= FEN/DTR	00001010
	PRITE (6,1015) FEND,PEXD	00001020
C		00001030
	30 IP (NORB.EQ.0) GO TO 900	00001040
C		00001050
C	*****	00001060
C		00001070
	WRIT2 (6,1018) RS,RSUN	00001080
	50 WRITE (6,1016)	00001090
	F1=0.D0	00001100
	CP= 2*PI/DFLOAT(NORB)	00001110
	DO 200 I=1,NORB	00001120
C	F1 IS ECC. LONG. H1,G1 ARE DUMMIES	00001130
	CALL ?CT (F1,0.D0,Z,H1,G1)	00001140
C	PRIBER CONE AHGLE	00001150
	BETA= (PI-DARCOS(CB))/DTR	00001160
C	THRUST CONE AHGLE	00001170
	TRETAD= (PI-THETA)/DTR	00001180
C	THRUST CLOCK ANGLE	00001190
	VEL(1)= X1DOT	00001200
	VEL(2)= Y1DOT	00001210
	VEL(3)= 0.D0	00001220

C E1 XNC E2 DO NOT HAVE TO BE UNIT VECTORS	00001230
CALL VCP(RS,VEL,E2)	00001240
CALL VCP(E2,RS,E1)	00001250
SCAV=0.DO	00001260
CCAV=0.DO	00007270
DO 100 J=1,3	00001280
SCAV= SCAV+U0(J)*E2(J)	00001290
700 CCAV= CCAV+U0(J)*E1(J)	00001300
PSIV= DATAN2(SCAV,CCAV)/DTR	00001310
FD= F1/DTR	00001320
C	00001330
C THRUST TO WEIGHT RATIO	00001340
TWR= A*PA*(X1*X1+Y1*Y1)	00001350
C	00001360
WRITE (6,1017) FD,X1,Y1,BETA,THETAD,PSIV,PA,TWR,U0	00001370
F1= F1+DF	00001380
C IF PLOT DATA DESIRED. PUT IT OUT	00001390
IF(IPL.NE 0) WRITE (8) FD,X1,Y1,BETA,THETAD,PSIV,PA,TWR,U0	00001400
200 CONTINUE	00001410
C	00001420
900 RETURN	00001430
C	00001440
	00007450
1001 FORMAT ('0 *****00001460	
1*****00001470	
1002 FORMAT (5H TIME,10X,10H TIME UNITS,15X,7H SECONDS,9X	00001480
1 ,5H HOURS,11X,4H DAYS,12X,1H)	00001490
1003 FORMAT (1P2D25.7,1P2D15.7,I9/)	00001500
1004 FORHAT (' THE EQUINOCTIAL ORBITAL ELEMENTS ARE')	00001510
1005 PORKBT (1P5D22.12/)	00001520
1006 FORMAT (16H0 THE COSTATE IS)	00001530
007 FORHAT (32H0 THE DERIVATIVE OF THE STATE IS)	00001540
008 FORHAT (34H0 THE DERIVATIVE OF THE COSTATE IS)	00001550
1009 FORMAT (1H0,7X,11H HAMILTONIAN,8X,12H PERIOD (HRS),5X,	00001560
1 15H PERICENTER (KM),7X,14H APOCENTER (KM),5X,14H CHAR ACC/RS**2	00001580
2 ,4X,17H C3 (KM**2/SEC**2))	
1010 PORKAT (2F20.12,1P4020.1 0 /)	00001590
1013 FORMAT (24H0 TIME SPENT IN SHADOW =, F12.6,27H MIN, FRACTION OF P	00001600
PERIOD =, F12.8)	00001610
1014 FORMAT (' THE CLASSICAL O.E.')	00001620
1015 FORHAT (1680 ENTRY ANGLE = ,F15.8,14H EXIT ANGLE = ,F15.8)	00001630
1016 FORMAT ('0 ECC. LONG. X1 Y1 PRIMER	00001640
1CONE A THRUST CONE A PSIV ACC FACT T/W RATIO'	00001650
2 /50X,'UF',13X,'UG',13X,'UW')	00001660
1017 FORMAT (1H0,1P8D15.5/46X,1P5D15.5/46X,1P5D15.5/46X,1P5D15.5)	00001670
7018 FORMAT ('0 SUN DIRECTION AND DISTANCE: ,1P4D15.5)	00001630
END	00001690

Subroutine OUTPC (OUTPCSS)

Description:

After the iteration, this subprogram prints a summary of the converged iteration parameters, t_f , etc.

Common Areas:

XMMM/ZL0(5), STEP(6), ZERF(6)

Z/ZF(10), DZ(10)

T/TF, TO

UNITS/UTS, UTM, UTH, UTD, UTKM, DTR, UTKG, UTKW, UTMS2

ELEM/ZP0(5), ZPF(5)

WF/WF(5)

TC/NOP

Called by:

CCNTL

ORIGINAL PAGE IS
OF POOR QUALITY

C	OUTPC/OUTPCSS	00000010
		00000020
J	TBIS SUBPROGRAM WRITES THE VALUES FOR THE FINAL CONVERGED	00000030
C	TRAJECTORY. IT IS CALLED BY THE MAIN PROGRAM CONTL-	00000040
C	SOLAR SAIL--ORBIT RAISING AND ESCAPE	00000050
C		00000060
C		00000070
C		00000080
	SUBROUTINE OUTPC	00090090
C		00000100
	IMPLICIT REAL*8 (A-H,O-Z) , INTEGER (I-N)	00000110
C		00000120
	COMMON /XMM/ZLO (5), STEP (6), ZERF (6)	00000130
	COMMON /Z/ZF (10), DZ (10)	00000140
	COMMON /T/TF, TO	00000150
	COMMON /UNITS/UTS, UTM, UTH, UTD, UTKM, DTR, UTKG, UTKW, UTMS2	00000160
	COMMON /ELEM/ZPO (5), ZPP (5)	00000170
	COMMON /WF/WF (5)	00000180
	COMMON /CCOM/CD (4)	00000190
	COMMON /TC/NOP	00000200
C		00000210
	DIMENSION DELZF (5), DELWF (5), WFC (5)	00000220
C		00000230
	WFC (1) = ZF (1) * UTKM	00000240
	WFC (2) = 0.0D0	00000250
	DUMMY = ZF (2) **2 + ZF (3) **2	00000260
	IF (DUMMY .GT. 1.0D-40) WFC (2) = DSQRT (DUMMY)	00000270
	WFC (3) = 0.0D0	00000280
	DUMMY = ZF (4) **2 + ZF (5) **2	00000290
	IF (DUMMY .GT. 1.0D-40) WFC (3) = 2.0D0 * DATAN (DSQRT (DUMMY)) / DTR	00000300
	WFC (4) = 0.0D0	00000310
	IF ((DABS (ZF (4)) .GT. 1.D-8) .AND. (DABS (ZF (5)) .GT. 1.D-6))	00000320
	1 WFC (4) = DATAN2 (ZF (4), ZF (5)) / DTR	00000330
	WFC (5) = 0.0D0	00000340
	IF ((DABS (ZF (2)) .GT. 1.D-8) .AND. (DABS (ZF (3)) .GT. 1.D-8))	00000350
	1 RPC (5) = DATAN2 (ZF (2), ZF (3)) / DTR	00000360
	PFC (5) = WFC (5) - WFC (4)	00000370
	DO 10 I = 1, 5	00000380
	DELWF (I) = WFC (I) - WF (I)	00000390
10	DELZF (I) = ZF (I) - ZPF (I)	00000400
	TF2 = TF * UTD	00000410
	TF1 = TF * UTS	00000420
C		00000430
	RRITE (6, 3000)	00000440
	RRITB (6, 3001)	00000450
	URITE (6, 3002) WFC	00000460
	WRITE (6, 3003)	00000470
	WRITE (6, 3002) (ZF (I), I = 1, 5)	00000480
	RRITE (6, 3004)	00000490
	WRITE (6, 3001)	00000500
	IF (NOP.EQ.1) WRITE (6, 3002) DELUF	00000510
	IF (NOP.EQ.2) WRITE (6, 3002) (DELWF (I), I = 1, 3)	00000520
	IF (NOP.EQ.3) WRITE (6, 3002) DELWF (1)	00000530
	GO TO (20, 30, 40), NOP	00000540
C		00000550
20	YRITE (6, 3003)	00000560
	WRITE (6, 3002) DELZF	00000570
	GO TO 100	00000580
C		00000590
30	DELZF (2) = DSQRT (ZF (2) **2 + ZF (3) **2) - ZPF (2)	00000600
	DELZF (3) = DSQRT (ZF (4) **2 + ZF (5) **2) - ZPF (3)	00000610

	WRITE (6,3011)	00000620
	WRITE (6,3002) (DELZF(I),I=1,3)	00000630
	GO TO 100	00000640
C		00000650
40	WRITE (6,3003)	00000660
	WRITE (6,3002) DELZF(1)	00000670
C		00000680
100	WRITE (6,3006)	00000690
	WRITE (6,3002) ZLO	00000700
	WRITE (6,3008)	00000710
	WRITE (6,3009) TF2,TF1,TF	00000720
	RETURN	00000730
3000	FORMAT (35H0 ACTUAL PINBL ORBITAL ELEMENTS ARE)	00000740
3001	FORHAT (1H0,10X,6RA (KM),18X,1HE,20X,7HI (DEG),10X,18HLON ASC NODE	00000750
	1 (DZG),6X,15HARG PERIC (DEG))	00000760
3002	FORHAT (1P5D23.15)	00000770
3003	FORMAT (1H0,5X,14HA (PLBNET RAD),15X,1HH,22X,1HK,22X,1HP,22X,1HQ)	00000780
3004	FORMAT (51H0 THE ERROR IN THE FINAL O.E. IS (ACTUAL - DESIRED))	00000790
3006	FORMAT (46H0 THE CONVERGED INITIAL GUESSED PARAMETERS ARE)	00000800
3008	FORMAT (29H0 THE MINIMIZED FINAL TIME IS)	00000810
3009	FORMAT (1H,10X,1PD22.15,7H DAYS =,D22.15,10H SECONDS =,1PD22.15,6	00000820
	1H UNITS)	00000830
3011	FORMAT (1H0,5X,14HA (PLANET RAD),8X,15HSQRT(H**2+K**2),8X,	00000840
	1 15HSQRT(P**2+Q**2))	00000850
	END	00000860

Subroutine PLANET

Description :

Sets various values associated with the planet including orbital elements, gravitational constant, radius, oblateness coefficient, rotation matrices to transform to ecliptic or equatorial coordinate systems.

Argument List:

GM Gravitational constant

Common Areas:

JD/TL

TERRA/AE, EC, W, ENE, AN, ECLMTX(3,3), EQUMTX(3,3)

UNITS/UTS, UTM, UTH, UTD, UTRM, DTR, UTKG, UTKW, UTMS2

PLNTS/IPNUM

J2/AJ2

```

C PLANET 00000010
C 00000020
C 00000030
C THIS SUBPROGRAM SETS THE VALUES FOR A PLANET'S ORBITAL ELEMENTS 00000040
C AND CALCULATES THE MEAN ANOMALY AT THE INITIAL TIME 00000050
C O.E. TAKEN FROM BATTIN, 1964. EPOCH 1960 JAN. 15, JD=2436935. 00000060
C CALCULATES COORDINATE TRANSFORMATION MATRICES AND SETS GRAVITY 00000070
C CONSTANT AND PLANET RADIUS. 00000080
C INPUT 00000090
C IPNUM--PLANET NUMBER 1-MERCURY,2-VENUS,3-EARTH,4-EARS 00000100
C TL--INITIAL TIME, BEGINNING OF LOW THRUST TBAJ 00000110
2 OUTPUT 00000120
C AE--EARTH'S SEMI-MAJOR AXIS 00000130
C EX--EARTH'S ECCENTRICITY 00000140
C W--ARGUMENT OF PERIHELION 00000150
C ENE--MEAN ORBITAL MOTION 00000160
C AN--MEAN ANOMALY AT TL 00000170
C ECLMTX--MATRIX FOR CONVERSION TO EARTH'S ECLIPTIC COORD FRAME 00000180
C EQUMTX--MATRIX FOR CONVERSION TO EQUATORIAL FRAME 00000190
C IMPLEMENTED FOR EARTH ONLY 03000200
C GM--PLANET GRAVITATIONAL CONSTANT 00000210
C UTKM--RADIUS 00000220
C AJ2--ECCENTRICITY 00000230
C 00000240
C 00000250
C 00000260
C SUBROUTINE PLANET(GM) 00000270
C 00000280
C IMPLICIT REAL*8(A-H,O-Z) 00000290
C 00000300
C COMMON /JD/ TL 00000310
C COMMON /TERRA/ AE,EC,W,ENE,AN,ECLMTX(3,3),EQUMTX(3,3) 00000320
C COMMON /UNITS/ UTS,UTM,UTH,UTD,UTKM,DTR,UTKW,UTMS2 00000330
C COMMON /PLNTS/ IPNUH 00000340
C COMMON /J2/ AJ2 00000350
C 00000360
C REAL*8 SEMIAX(4),ECCEN(4),MLP(4),MORBM(4),GRAVC(4),RAD(4) 00000370
C REAL*8 J2(4),MLE(4),ANGOBL(4),LNODE(4),INCL(4) 00000380
C 00000390
C DATA STATEMENTS 00000400
C DATA SEMIAX/0.387099D0,0.723322D0,1.D0,1.523691D0/ 00000410
C DATA ECCEN/0.205627D0,6.793D-3,1.6726D-2,9.3368D-2/ 00000420
C DATA MLP/76.83309D0,131.00931D0,102.25253D0,335.32269D0/ 00000430
C DATA MORBM/4.092339D0,1.602131D0,0.985609D0,0.524033D0/ 00000440
C DATA GRAVC/22181.6D0,324860.1D0,398601.2D0,42828.4D0/ 00000450
C DATA RAD/2435.D0,6052.D0,6378.16D0,3393.4D0/ 00000460
C DATA J2/0.D0,0.D0,1.0827D-3,0.D0/ 00000470
C DATA MLE/222.62165D0,174.29431D0,100.15815D0,258.76729D0/ 00000480
C DATA ANGOBL/0.D0,0.D0,23.45D0,0.D0/ 00000490
C DATA LNODE/47.85714D0,76.31972D0,0.D0,Y9.24903D0/ 00000500
C DATA INCL/7.00399D0,3.39423D0,0.D0,1.84991D0/ 00000510
C 00000520
C 00000530
C SEMI-MAJOR AXIS 03000540
C AE=SEMIAX(IPNUH) 00000550
C ECCENTRICITY 00000560
C EC=ECCEN(IPNUM) 00000570
C ARGUMENT OF PERIHELION 00000580
C W=MLP(IPNUM)-LNODE(IPNUM) 00000590
C MEAN ORBITAL MOTION 00000600
C ENE=MORBM(IPNUM) 00000610

```

GH(KM**3S**2)	00000620
GM=GRAVC(IPNUH)	00000630
C RADIUS (KH)	00000640
UTKM=RAC(IPNUH)	00000650
C 52	00000660
AJ2=J2(IPNUH)	00000670
C	00000680
C MEAN ANOMALY AT EPOCH	00000690
AN= MLE(IPNUM)-MLP(IPNUM)	00000700
C MEAN ANOMALY AT TIKE TL	00000710
AN= AN+ENE*(TL-2436935.DO)	00000720
AN=AN/360.DO	00000730
AN=AN-IDINT(AN)	00000740
C	00000750
AN= AN*360.DO*DTR	00000760
W = W*DTR	00000770
C ENE WILL BE HULTIPLIED BY UTD IN INPUTSS,SINCE UTD DEPENDS	00000780
C ON UTKM,WHICH CAN BE MODIFIED BY THE INPUT STREAM	00000790
ENE= ENE*DTR	00000800
C	00000810
C SET UP ARRAY FOR USE IN CONVERTING TO EARTH'S ECLIPTIC FRAHE	00000820
C	00000930
DCL=DCOS(LNODE(IPNUM)*DTR)	00000840
DSL=DSIN(LNODE(IPNUM)*DTR)	00000850
DCI=DCOS(INCL(IPNUM)*DTR)	00000860
DSI=DSIN(INCL(IPNUH)*DTR)	00000870
C	00000880
ECLMTX(1,1)=DCL	00000890
ECLMTX(2,1)=-DCI*DSL	00000900
ECLHTX(3,1)=DSI*DSL	00000910
ECLMTX(1,2)=DSL	00000920
ECLMTX(2,2)=DCI*DCL	00000930
ECLMTX(3,2)=-DSI*DCL	00000940
ECLMTX(1,3)=0.DO	00000950
ECLMTX(2,3)=DSI	00000960
ECLMTX(3,3)=DCI	00000970
C	00000980
C	00000990
C SET UP EQUmtx FOR USE IN CONVERSION TO EQUATORIAL FRAME	00001000
C	00001010
C CURRENTLY IMPLEMENTED ONLY FOR EARTH	00001020
EQUmtx(1,1)=1.DO	00001030
EQUmtx(2,1)=0.DO	0F001040
EQUmtx(3,1)=0.DO	00001050
EQUmtx(1,2)=0.DO	00001060
EQUmtx(1,3)=0.DO	00001070
IF (IPNUM.NE.3) GO TO 5	00001080
DUM=ANGOBL(IPNUH)*DTR	00001090
DCOSD=DCOS(DUM)	00001100
DSIND=DSIN(DUM)	00001110
EQUmtx(2,2)=DCOSD	00001120
EQUmtx(3,2)=-DSIND	00001130
EQUmtx(2,3)=DSIND	00001140
EQUmtx(3,3)=DCOSD	00001150
GO TO 10	00001160
C	00001170
5	00001180
EQUmtx(2,2)=1.DO	00001190
EQUmtx(3,2)=0.DO	00001200
EQUmtx(2,3)=0.DO	00001210
EQUmtx(3,3)=1.DO	00001220
10	00001230
C	00001240
RETURN	00001250
END	

ORIGINAL PAGE 15
OF POOR QUALITY

Subroutine PRTN (PRTNSS)

Description:

Prints information about the Newton iteration.

Argument List:

KOUNT, NOI

KOUNT The number of trajectory calls

NOI The number of iterations

Common Areas:

XMMM/X(5), XS(6), Y(6)

T/TE, TO

Called by :

ITER

PRTN/PRTNSS	00000010
C	00000020
C THIS PROGRAM IS CALLED BY THE ITERATOR AND PRINTS	00000030
C	00000050
C	00000060
SUBROUTINE PRTN(KOUNT,NCI)	00000070
C	00000080
IMPLICIT REAL*8(A-H,O-Z)	00000090
COMMON /XMMH/X(5),XS(6),Y(6)	00000100
COMMON /T/TF,T0	00000110
C	00000120
N=5	00000130
M=6	00000140
URITE (6,1000)	00000150
WRITE (6,1001) NCI,KOUNT	00000160
YRITE (6,1002)	00000170
WRITE (6,1003) (X(J),J=1,N)	00000180
FRITE (6,1004)	00000190
FRITE (6,1003) (Y(J),J=1,M)	00000200
BRITE (6,1005) TF	00000210
1000 FORMAT (29H0 ITER NO. 'IRBJECTOBY CALLS)	03000220
1001 FORHAT (1H0,I6,5X,I6)	00000230
1002 FORHAT (2H0X)	00000240
1003 FORMAT (1X,1P5D23.15)	00000250
1004 FORMAT (2H0Y)	00000260
1005 FORMAT (5H0 TF=,1PD26.16)	00000270
RETURN	00000280
END	00000290

Subroutine QUAD (QUAD4, QUAD8, QUAD16; QUAD32)

Description:

These are 4, 8, 16, and 32 point vector Gaussian quadratures used for the averaging.

Argument List:

XL, XU, FCT, Y, Z, G, H, N

XL Lower bound

XU Upper bound

FCT Dummy program name - called to evaluate integrand

Y The resultant integrated function

Z . Orbital elements and adjoints

G Dummy variable

H Dummy variable

N Dimension

Called by:

FUNCT

Calls Subroutines:

FCT

C	QUAD	00000010
C		Or1000020
C	THIS IS A MODIFIED QUADRATURE PROGRAM FOR VECTOR VALUED FUNCTIONS.	00000030
C	COMPUTES INTEGRAL OF THE FUNCTION G (OR H) OQSR X FROM XL TO XU. THE	00000040
C	RESULT IS Y. BOUTINE USES A 4 POINT GAUSS QUADRATURE.	00000050
C		00900060
C		00000070
C		00000080
C	SUBROUTINE QUAD(XL,XU,FCT,Y,Z,G,H,N)	00000090
C		00000100
C	IMPLICIT REAL*8(A-H,O-Z), INTEGER(I-N)	00000110
C	DIMENSION Y(1),H(1),G(1),Z(1)	00000120
C		00000130
	A= .5D0*(XU+XL)	00000140
	B= XU-XL	00000150
	C= .43056815579702629D0*B	00000160
	K=1	00000170
	GO TO 50	00000180
10	DO 20 I=1,N	00000190
20	Y(I)= .17392742256872693D0*G(I)	00000200
	C= .16999052179242813D0*B	00000210
	K=2	00000220
	GO TO 50	00000230
30	DO 40 I=1,N	00000240
40	Y(I)= B*(Y(I) + .32607257743127307D0*G(I))	00000250
	RETURN	00000260
50	CALL FCT(A-C,A+C,Z,H,G)	00000270
	DO 60 I=1,N	00000280
60	G(I)=G(I) + H(I)	00000290
	GO TO (10,30), K	00000300
	END	00000310

C	QUAD/QUAD8	00000010
C		00000020
C	THIS IS A MODIFIED QUADRATURE PROGRAM FOR VECTOR VALUED FUNCTIONS.	00000030
C	COMPUTES THE INTEGRAL OF THE FUNCTION G (OR H) OVER X FROM XL TO XU.	00000040
C	THE RESULT IS Y. ROUTINE USES A 8 POINT GAUSS QUADRATURE.	00000050
C		00000060
C		00000070
C		00000080
	SUBROUTINE QUAD(XL,XU,FCT,Y,Z,G,H,N)	00000090
C	IMPLICIT REAL*8(A-H,O-Z)	00000100
	DIMENSION Y(1),H(1),G(1),Z(1)	00000110
C		00000120
	A=.5D0*(XU+XL)	00300130
	B=XU-XL	00000140
	C=.48014492824876812D0*B	00000150
	K=7	00000160
	GO TO 500	00000170
10	DO 20 I=1,N	00000180
20	Y(I)=.50614268145188130D-1*G(I)	00000190
	C=.39833323870681337D0*B	00000200
	K=2	00500210
	GO TO 500	00000220
30	DO 40 I=1,N	00000230
40	Y(I)=Y(I)+.11119051722668724D0*G(I)	00000240
	C=.26276620495816449D0*B	00000250
	K=3	00000260
	GO TO 500	00000270
50	DO 60 I=1,N	00000280
60	Y(I)=Y(I)+.15685332293894364D0*G(I)	00000290
	C=.9171732124782490D-1*B	00000300
	K=4	00000310
	GO TO 500	00000320
70	DO 80 I=1,N	00000330
e5	Y(I)=B*(Y(I)+.18134189168918099D0*G(I))	00000340
	RETURN	00000350
500	CALL FCT(A-C,A+C,Z,H,G)	00000360
	DO 510 I=1,N	00000370
510	G(I)=G(I)+H(I)	00000380
	GO TO (10,30,50,70),K	00000390
	END	00000400
		00000410

QUAD16

C THIS IS A MODIFIED QUADREATURE INTEGRATION PROGRAM FOR
C VECTOR VALUED FUNCTIONS FO OHE VARIABLE. IT INTEGRATES
C G (OR H) OVER X FROX XL TO XU. THE RESULT IS Y.
C EVALUATION IS BY A 16 POINT GBUSS QUADRATURZ FORMULA.

```

C      SUBROUTINE QUAD(XL,XU,PCT,Y,Z,G,H,N)
C
C      IMPLICIT REAL*8(A-H,O-Z)
C      DIMENSION Y(1),Z(1),G(1),H(1)
C
C      A= .5D0*(XU+XL)
C      B= XU-XL
C      C= .49470046749582497C0+B
C      K=1
C      GO TO 500
10    DO 20 I=1,N
20      Y(I)= .13576229705877047D-1*G(I)
C      C= .47228751153661629D0*B
C      K=2
C      GO TO 500
30    DO 40 I=1,N
40      Y(I)= Y(I) + .31126761969323946D-1*G(I)
C      C= -43281560119391587D0*B
C      K=3
C      GO TO 500
50    DO 60 I=1,N
60      Y(I)= Y(I) + .47579255841246392D-1*G(I)
C      C= .37770220417750152D0*B
C      K=4
C      GO TO 500
70    DO 80 I=1,N
80      Y(I)= Y(I) + .62314485627766936D-1*G(I)
C      C= .30893812220132187D0*B
C      K=5
C      GO TO 500
90    DO 100 I=1,N
100     Y(I)=Y(I) + .7479799440828837D-1*G(I)
C      C= .22900838882861369D0*B
C      K=6
C      GO TO 500
110   DO 120 I=1,N
120     Y(I)= Y(I) + .8457825969750127D-1*G(I)
C      C= .14080177538962946D0*B
C      K=7
C      GO TO 500
130   DO 140 I=1,N
140     Y(I)= Y(I) + .9130170752246179D-1*G(I)
C      C= .47506254918818720D-1*B
C      K=8
C      GO TO 500
150   DO 160 I=1,N
160     Y(I)= B*(Y(I) + .9472530522753425D-1*G(I))
C      RETURN
500   CALL FCT(A-C,A+C,Z,H,G)
C      DO 510 I=1,N
510     G(I)= G(I) + H(I)
C      GO TO (10,30,50,70,90,110,130,150),K
C      STOP
C      END

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000200
00000250
00000260
00000270
00000280
00000230
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000605
00000610

C	QUAD32	00000010
C		00000020
C	THIS IS A MODIFIED QUADRATURE INTEGRATION PROGRAH POR	00000030
C	VECTOR VALUED FULCTIONS OF ONE VARIABLE. IT INTEGRATES	00000040
C	G (OR H) OVER X FROM XL TO XU. THE RESULT IS Y.	00000050
C	EVALUATION IS BY A 32 POINT GAUSS QUADRATURE FORMULA.	00000060
C		00000070
C	SUBROUTINE QUAD(XL,XU,FCT,Y,Z,G,H,N)	00000080
C		00000090
	IMPLICIT REAL*8(A-H,O-Z)	00000100
	DIMENSION Y(1),Z(1),G(1),H(1)	00000110
C		00000120
	A= .5D0*(XU-XL)	00000130
	E= XU-XL	00000140
	C= .49863193092474078D0*B	00000150
	K=1	00000160
	GO TO 500	00000170
10	DO 20 I=1,N	00000180
20	Y(I)= .35093050047350483D-2*G(I)	00000190
	C= .49280575577263417D0*B	00000200
	K=2	00000210
	GO TO 500	00000220
30	DO 40 I=1,N	00000230
40	Y(I)= Y(I) + .8137197365452835D-2*G(I)	00000290
	C= .48238112779375322D0*B	00000250
	K=3	00000260
	GO TO 500	00000270
50	DO 60 I=1,N	00000280
60	Y(I)= Y(I) + .12596032654631030D-1*G(I)	00000290
	C= .46745303796886984D0*B	00000300
	K=4	00000310
	GO TO 500	00000320
70	DO 80 I=1,N	00000330
80	Y(I)= Y(I) + .17136931456510717D-1*G(I)	00000340
	C= .44816057788302606D0*B	00000350
	K=5	00000360
	GO TO 500	00000370
90	DO 100 I=1,N	00000380
100	Y(I)= Y(I) + .21417949011113340D-1*G(I)	00000390
	C= .42468380686628499D0*B	00000400
	K=6	00000410
	GO TO 500	00000420
110	DO 120 I=1,N	00000430
120	Y(I)= Y(I) + .25499029631188088D-1*G(I)	00000440
	C= .39724189798397120D0*B	00000450
	K=7	00000460
	GO TO 500	00000470
130	DO 140 I=1,N	00000480
140	Y(I)= Y(I) + .29342046739267774D-1*G(I)	00000490
	C= .36609105937014484D0*B	00000500
	K=8	00000510
	GO TO 500	00000520
150	DO 160 I=1,N	00000530
160	Y(I)= Y(I) + .32911111389180923D-1*G(I)	00000540
	C= .33152213346510760D0*B	00000550
	K=9	09000560
	GO TO 590	00000570
170	DO 180 I=1,N	00000580
180	Y(I)= Y(I) + .36172897054424253D-1*G(I)	00000590
	C= .29385787862038116D0*B	00000600
	K=10	00000610

	GC TO 500	00000620
150	DO 200 I=1,N	00000630
200	Y(I)= Y(I) + .39096947893535153D-1*G(I)	00000640
	C= .25344995446611470D0*B	00000650
	K=11	00000660
	GO TO 503	00000670
210	DO 220 I=1,N	00000680
220	Y(I)= Y(I) + .41655962713473378D-1*G(I)	00000690
	C= .21067563806531767D0*B	00000700
	K=12	00000770
	GO TO 500	00000720
230	DO 240 I=1,N	00000730
240	Y(I)= Y(I) + .43826046502201906D-1*G(I)	00000740
	C= .16593430114106382D0*B	00000750
	K=13	00000760
	GO TO 500	00000770
250	DO 260 I=1,N	00000780
260	Y(I)= Y(I) + .45596939347881942D-1*G(I)	00000790
	C= .11964368112606854D0*B	00000800
	K=14	00000810
	GO TO 500	00000820
270	DO 280 I=1,N	00000830
280	Y(I)= Y(I) + .46922199540402283D-1*G(I)	00000840
	C= .7223598079139825D-1*B	00000850
	K=15	00000860
	GO TO 500	00000870
290	DO 300 I=1,N	00000860
300	Y(I)= Y(I) + .47819360039637430D-1*G(I)	00000890
	C= .24153832843869158D-1*B	00000900
	K=16	00000910
	GO TO 500	00000920
310	DO 320 I=1,N	00000930
320	Y(I)= B*(Y(I) + .48270044257363900D-1*G(I))	00000940
	RETURN	00000950
500	CALL FCT(A-C,A+C,Z,H,G)	00000960
	DO 510 I=1,N	00000970
510	G(I)= G(I) + H(I)	00000980
	GO TO (10,30,50,70,90,110,130,150,170,190,210,230,250,270,290,310)	00000990
	END	00001010

Subroutine RK4

Description:

This is 4 point Runge-Kutta integrator which is just the IBM Scientific Subroutine Package version without the accuracy checks. (4)

Argument List:

PRMT, Y, DERV, NDIM, IHLF, FCT, OUTP, AUX

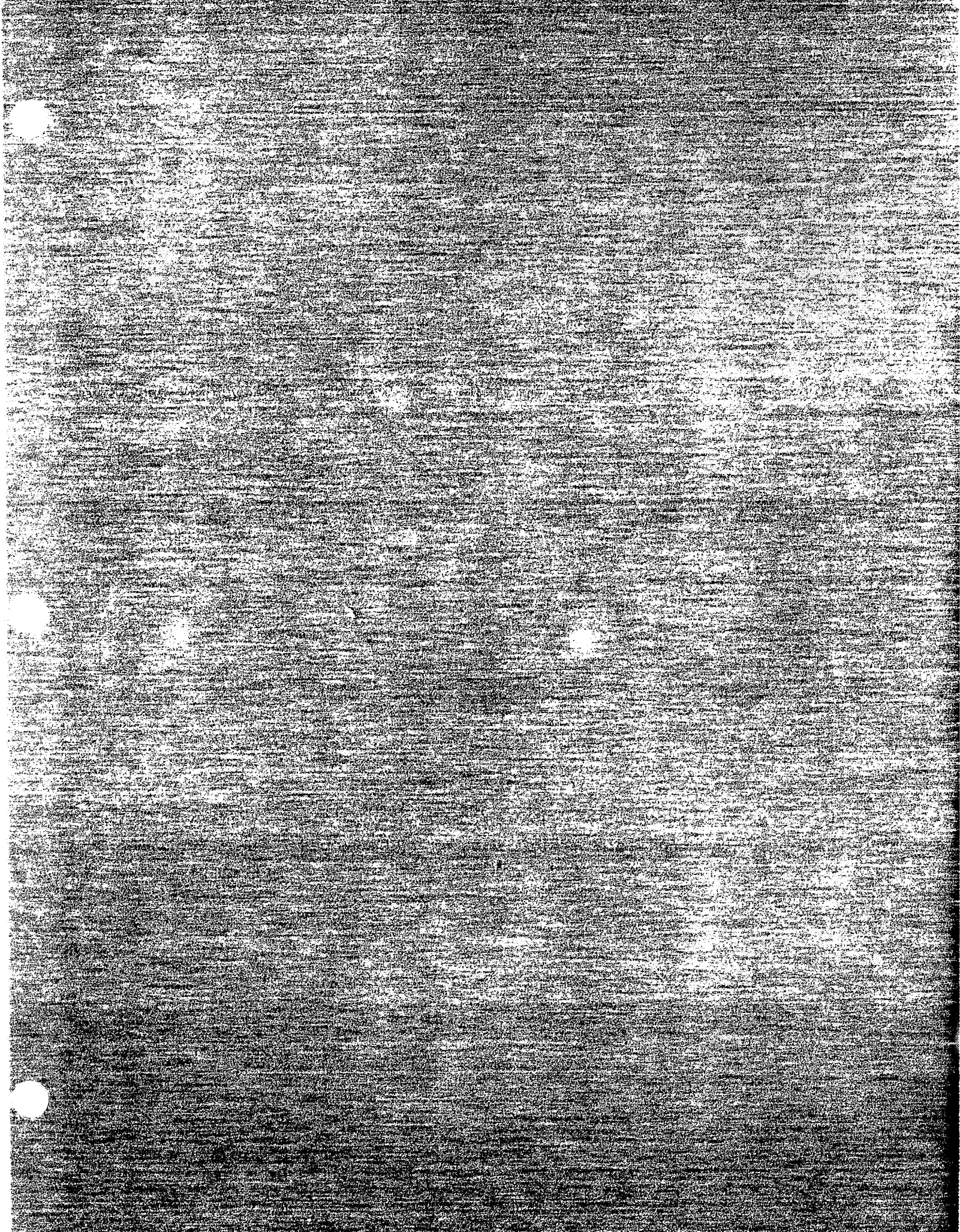
PRMT(5)	Initial time measured from launch time, time of flight, integration step size, final 2 not used.
Y	State and costate (initial as input, final as output)
DERV	Final derivative
NDIM	Dimension of Y
IHLF	Flag = 12 if TF = TO = 13 if DT(TF - TO) < 0
FCT	Dummy subroutine name - called to evaluate derivative
GUT?	Dummy output subroutine name
AUX	Dummy array used for intermediate calculation

Called by :

TRAJ

ORIGINAL PAGE IS
OF POOR QUALITY

C	RK4	0000001C
		00000020
	THIS IS A RUNGE KUTTA INTEGRATOR OF 4TH ORDER-	00000030
		00000040
C		00000050
	SUBROUTINE RK4 (PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX)	00000060
C		00000070
	IMPLICIT REAL*8(A-H,O-Z)	00000080
	DIMENSION Y(1),DERY(1),AUX(1,1),A(4),B(4),C(4),PRMT(1)	00000090
C		00000100
	X=PRMT(1)	00000110
	XEND=PRMT(2)	00000120
	H=PRMT(3)	00000130
	PRMT(5)=0.D0	00000140
	IREC=0	00000150
	IHLF=0	00000160
	CALL FCT(X,Y,DERY)	00000170
C		00000180
C	ERROR TEST	00000190
	IF (H*(XEND-X))38,37,2	00000200
C		00000210
C	PREPARATION FOR R-K METHOD	00000220
2	A(1)=.SD0	00000230
	A(2)=.29289321881345248D0	00000240
	A(3)=1.7071067811865475D0	00000250
	A(4)=.16666666666666667D0	00000260
	B(1)=2.D0	00000270
	B(2)=1.D0	00000280
	B(3)=1.D0	00000290
	B(4)=2.D0	00000300
	C(1)=.5D0	00000310
	C(2)=A(2)	00000320
	C(3)=A(3)	00000330
	C(4)=.5D0	00000340
C		00000350
C	PREPARATIONS OF FIRST R-K STEP	00000360
	DO 3 I=1,NDIM	00000370
3	AUX(1,I)=0. DO	00000380
	IEND=0	00000390
C		00000400
C		00000410
C	START OF A R-K STEP	00000420
4	IS=0	00000430
	IF ((X+2.D0*H-XEND)*H) 7,6,5	00000440
5	H=(XEND-X)/2.D0	00000450
6	IEND=1	00000460
7	CALL OUTP(X,Y,DERY,IREC,NDIM,PRMT)	00000470
	IF (PRMT(5))40,8,40	00000480
C		00000490
C		00000500
C	START OF R-K LOOP	00000510
8	J=1	00000520
10	AJ=A(J)	00000530
	BJ=B(J)	00000540
	CJ=C(J)	00000550
	DO 11 I=1,NDIM	00000560
	R1=H*DERY(I)	00000570
	R2=AJ*(R1-BJ*AUX(1,I))	00000580
	Y(I)=Y(I)+R2	00000590
	R2=R2+R2+R2	00000600
11	AUX(1,I)=AUX(1,I)+R2-CJ*R1	00000610



Subroutine **SHADOW** (SHADOWSS)

Description:

Sets up the shadow quartic equation (Appendix D, Reference 1), solves it, checks the roots to find the entrance and **exit** of the orbit from shadow if there is intersection and calculates $\frac{df}{dz}$ (Eq. D.5) for entry and exit.

Argument List:

Z

Z Orbital elements and their adjoints

Common Areas:

SOL/XSUN, YSUN, ZSUN, RSUN

SHAD/EEN, **E**, DFEN(5), DFEX(5), ISHAD

Called by:

FUNCT, OUTP

PRECEDING PAGE BLANK NOT FILLED

```

C SHADOW/SHADOWSS                                00000010
C                                                    00000020
C THIS PROGRAY DETERMINES IF h GIVEN ORBIT PASSES THROUGH 00000030
C THE EARTH'S SHADOW AND IF SO WHAT THE ENTRY AND EXIT 00000040
C ANGLES ARE. IT ALSO CALCULATES THE PARTIAL DERIVITIVE 00000050
C VECTOR OF F WRT THE O.E. AT THE ENTRY AND EXIT POINTS. 00000060
C INPUT AND OUTPUT IN EQUINOCTIAL COORD, 00000070
C INPUT 00000080
C      Z-- 10 VECTOR OF O.E. AND COSTATE (COSTATE NOT USED) 00000090
C      XSUN,YSUN,ZSUN--SUN'S DIRECTION IN EQ. COORD. (UNIT VECTOR) 00000100
C OUTPUT 00000110
C      FEN--ENTRY AHGLE 00000120
C      FEX--EXIT ANGLE 00000130
C      DFEN--DERIVATIVE OF P AT ENTRY 00000140
C      DFEX--DERIVATIVE OF F AT EXIT 00000150
C      ISHAD--FLAG=0 IF ORBIT NOT INTERSECT SHADOW 00000160
C              =2 IF ORBIT ENTER AND EXIT FROM SHADOW 00000170
C 00000180
C 00000190
C 00000200
C SUBROUTINE SHADOW(Z) 00000210
C 00000220
C      IMPLICIT REAL*8(A-H,O-Z) 00000230
C 00000240
C      COMMON /SOL/ XSUN,YSUN,ZSUN,RSUN 00000250
C      COMMON /SHAD/ FEN,FEX,DFEN(5),DFEX(5),ISHAD 00000260
C 00000270
C      DIMENSION DSDX(5),AP(4),RT(4),Z(10) 00000280
C 00000290
C      NAMELIST /DUMFEN/ FEN,DUM 00000300
C      NAMELIST /DUMFEX/ FEX,DUM 00000310
C      NAMELIST /EQ/ AP,RT,NRE 00000320
C      NAMELIST /EQ2/BETA,B1,B2,B3,D1,D2,D3,H1,H2,H3,G1,G2,A0 00000330
C      NAMELIST /PR4/ DUM,DSDP,DSDX 00000340
C      NAMELIST /PR5/ I,II,CP,SP,X1,Y1 00000350
C      NAMELIST /PR6/ I,II,EQN 00000360
C 00000370
C 00000380
C 00000390
C CALCULATE POLYNOMIAL COEF. 00000400
C 00000410
C      BETA= DSQRT(1.D0-Z(2)**2-Z(3)**2) 00000420
C      BETA= 1.D0/(1.D0+BETA) 00000430
C      B1= 1.D0-Z(2)**2*BETA 00000440
C      B2= Z(2)*Z(3)*BETA 00000450
C      B3= 1.D0-Z(3)**2*BETA 00000460
C      D1= 1.D0-XSUN**2 00000470
C      D2= 1.D0-YSUN**2 00000480
C      D3= 2.D0*XSUN*YSUN 00000490
C      C1=B2**2 00000500
C      C2=B3**2 00000510
C      C3=B2*B3 00000520
C      C4=B1*B2 00000530
C      H1=D1*(B1**2-C1)+D2*(C1-C2)-D3*(C4-C3) 00000540
C      H2=-2.D0*(D1*B1*Z(3)+D2*B2*Z(2))+D3*(B2*Z(3)+B1*Z(2)) 00000550
C      H3=D1*(C1+Z(3)**2)+D2*(C2+Z(2)**2)-D3*(C3+Z(2)*Z(3)) 00000560
C      1 -1.D0/Z(1)**2 00000570
C      G1= 2.D0*(D1*C4+D2*C3)-D3*(C1+B1*B3) 00000580
C      G2= -2.D0*(D1*B2*Z(3)+D2*B3*Z(2))+D3*(B3*Z(3)+B2*Z(2)) 00000590
C      C1= G1**2 00000600
C      C2= G2**2 00000610
C      C3= G1*G2

```

A0= H1**2+C1	00000620
C WRITE (6,EQ2)	00000630
AP(1)=2.D0*(H1*H2+C3)/A0	00000540
AP(2)= (H2**2+2.D0*H3*H1-C1+C2)/A0	00000650
AP(3)= 2.D0*(H3*H2-C3)/A0	00000660
AP(4)= (H3**2-C2)/A0	00000670
C	00000680
C CALL SUBROUTINE TO SOLVE A QUARTIC EQN.	00000690
C	00000700
CALL DQRTIC(AP,RT,NRE)	00000710
C	00000720
C WRITE (6,EQ)	00000730
C	00000740
C NRE= NUMBER OF REAL ROOTS, MUST BE EQUAL TO 0,2,OR 4	00000750
C ROOTS ARE RT(I), OR RT(1),RT(2),RT(3)+-RT(4)*I, OR	00000760
C RT(1)+-RT(2)*I,RT(3)+-RT(4)*I	00000770
IF ((NRE.EQ.1).OR.(NRE.EQ.3)) GO TO 130	00000780
C	00000790
FEN= 100.D0	00000800
PEX= 100.D0	00000810
ISHAD= 0	00000820
I=0	00000830
10 I=I+1	00000840
IF ((ISHAD.EQ.2).OR.(I.EQ.(NRE+1))) GO TO 120	00000850
C	00000860
CP= RT(I)	00000870
SF= DSQRT(1.D0-CP**2)	00000880
C	00000890
C HEMISPHERE CHECK	00000900
II=1	00000910
20 X1= B1*CP+B2*SF-Z(3)	00000920
Y1= B3*SF+B2*CP-Z(2)	00000930
C WRITE (6,PR5)	00000940
IF ((X1*XSUN+Y1*YSUN).LT.0.D0) GO TO 40	00000950
30 IF (II.EQ.2) GO TO 10	00000960
II=2	00000970
SF=-SF	00000980
GO TO 20	00000990
C	00001000
C IS SHADOW EQUATION ZERO?	00001010
40 EQN= D1*X1**2+D2*Y1**2-D3*X1*Y1-Z(1)**(-2)	00001020
C RRLTE (6,PR6)	00001030
IF (DABS(EQN).GT.1.D-6) GO TO 30	00001040
C	00001050
C ROOT HAS PASSED TESTS—NOW CHECK TO SEE IF EXIT OR ENTRY ANGLE	00001060
DXDF= -B1*SF+B2*CP	00001070
DYDF= -B2*SF+B3*CP	00001080
DSDF= (2.D0*D1*X1-D3*Y1)*DXDF +(2.D0*D2*Y1-D3*X1)*DYDF	00001090
DUM= DATAN2(SF,CP)	00001100
IF (DSDP) 70,50,60	00001110
C	00001120
C ORBIT IS TANGENT TO SHADOW	00001130
50 BRITE (6,1010)	00001140
GO TO 30	00001150
C	00001160
C IS PEX ALREADY FOUND?	00001170
60 IF (PEX.EQ.1.D2) GO TO 80	00001180
C YES	00001190
WRITE (6,DUMPEX)	00001200
GO TO 30	00001210
C	00001220

C IS FEN ALREADY POUND?	00001230
70 IF (FEN.EQ.1.D2) GO TO 80	00001240
C YES	00001250
WRITE (6,DUMFEN)	00001260
GO TO 30	00001270
C	00001280
C CALCULATE DSDX	00001290
80 ZETA= Z(3)*SF-Z(2)*CF	00001300
BETA3= BETA**3/(1.D0-BETA)	00001310
PZ5= Z(2)*BETA3	00001320
PZ6= Z(3)*BETA3	00001330
DXDH= -2.D0*Z(2)*BETA*CF+Z(3)*BETA*SF+PZ5*ZETA*Z(2)	00001340
DXDK= Z(2)*BETA*SF-1.D0+PZ6*Z(2)*ZETA	00001350
DYDH= Z(3)*BETA*CF-1.D0-PZ5*Z(3)*ZETA	00001360
DYDK= -2.D0*Z(3)*BETA*SF+Z(2)*BETA*CF-PZ6*Z(3)*ZETA	00001370
DSDX(1)= 2.D0*Z(1)**(-3)	00001380
DUM1= 2.D0*D1*X1-D3*Y1	00001390
DUM2= 2.D0*D2*Y1-D3*X1	00001400
DSDX(2)= DUM1*DXDH+DUM2*DYDH	00001410
DSDX(3)= DUM1*DXDK+DUM2*DYDK	00001420
D=2.D0/(1.D0+Z(4)**2+Z(5)**2)	00001430
DXSP= (-YSUN*Z(5)-ZSUN)*D	00001440
DXSQ= YSUN*Z(4)*D	00001450
DYSP= XSUN*Z(5)*D	00001460
DYSQ= (-XSUN*Z(4)+ZSUN)*D	00001470
DUM1= -2.D0*X1*(X1*XSUN+Y1*YSUN)	00001480
DUM2= -2.D0*Y1*(Y1*YSUN+X1*XSUN)	00001490
DSDX(4)= DUM1*DXSP+DUM2*DYSP	00001500
DSDX(5)= DUM1*DXSQ+DUM2*DYSQ	00001510
C WRITE (6,PR4)	00001520
ISHAD=ISHAD+1	00001530
IF (DSDX.LT.0.D0) GO TO 100	00001540
C	00001550
C EXIT ANGLE AND DERIVATIVE	00001560
FEX=DUM	00001570
DO 90 J=1,5	00001580
90 DFEX(J)= -DSDX(J)/DSDX	00001590
GO TO 10	00001600
e	00001610
C ENTRY ANGLE AND DERIVATXYE	00001620
100 FEN=DUM	00001630
DO 110 J=1,5	00001640
110 DFEN(J)= -DSDX(J)/DSDX	00001650
GO TO 10	00001660
C	00001670
C	00001680
120 IF ((ISHAD.EQ.0).OR.(ISHAD.EQ.2)) RETURN	00001690
WRITE (6,1020) ISRAD	00001700
ISHAD= 0	00001710
RETURN	00001720
C	00001730
130 WRITE (6,1030) NRE	00001740
STOP	00001750
C	00001760
C	00001770
1010 FORMAT (33H0 DSDX=0, ORBIT TANGENT TO SHADOW)	00001780
1020 FORMAT (15H0 ERROR--ISHAD=,14)	00001790
1030 FORMAT (49H0 QRTIC HAS RETURNED WITH NUMBER OF REAL ROOTS =,14)	00001800
END	00001810

Subroutine SUN (SUNSS)

Description:

Calculates planet-sun unit vector in the equinoctial frame and the distance in A.U.'s divided by planet's semimajor axis.

Argument List:

T, Z

T Time

Z State and costate

Common areas:

SOL/RS(3), R

TERRA/AE, EC, W, ENE, AN, ECLMTX(3,3), EQUMTX(3,3)

Called by :

FUNCT, OUTP

```

C SUN/SUNSS                                00900010
C                                           00900020
C                                           00000030
C THIS PROGRAM CALCULATES THE PLANET TO SUN DIRECTION AND 00000040
C DISTANCE FOR A GIVEN TIME, OUTPUT IN THE EQUINOCTIAL 00000050
C COORDINATE FRAME. USED WITH PLANETS.        00000060
C INPUT                                         00000070
C   Z--10 VECTOR C? EQ. OR. AND COSTATE(NOT USED) 00000080
C   AE--ORBIT SEMIMAJOR AXIS                   00000090
C   EC--ORBIT ECCENTRICITY                     00000100
C   W--ARG. OF PERIH.                         00000110
C   ENE--MEAN ORBITAL MOTION                  00000120
C   AN--MEAN ANOMALY AT BEGINNING OF TRAJECTORY (TO) 00000130
C   ECLMTX--MATRIX FOR CONVERSION TO ECLIPTIC FRAME 00000149
C   EQUMTX--MATRIX FOR CONVERSION TO EQUATORIAL FRAME 00000150
C   T--PRESENT TIME                           00000160
C OUTPUT                                        00000170
C   RS--UNIT VECTOR FROM PLANET TO SUN, EQUINOCTIAL COORD. 00000180
C   R--DISTANCE FROM PLANET TO SUN AT TIME T    00000190
C                                           00000200
C                                           00000210
C                                           00000220
C   SUBROUTINE SUN(T,Z)                       00000230
C                                           00000240
C   IMPLICIT REAL*8(A-H,O-Z)                 00000250
C                                           00000260
C   COMMON /SOL/ RS(3), R                    00000270
C   COMMON/TERRA/ AE,EC,W,ENE,AN,ECLMTX(3,3),EQUMTX(3,3) 00000280
C                                           00000290
C   DIMENSION RS1(3),CM(3,3),Z(10),RS2(3),TMP(3) 00000300
C                                           00000310
C                                           00000320
C MEAN ANOMALY AT TIME T                     00000330
C   AA= AN+ENE*T                             00000340
C                                           00000350
C TRUE ANOMALY--CORRECT THRU ECCENTRICITY CUBED 00000360
C   F=AA+(2.D0*EC-.25D0*EC**3)*DSIN(AA)+1.25D0*EC**2*DSIN(2.D0*AA) 00000370
C   1 +1.08333333333333D0*EC**3*DSIN(3.D0*AA) 00000380
C   B=F+W                                     00000390
C                                           00000400
C DISTANCE BETWEEN PLANET AND SUN IN PLANET RADII 00000410
C   R=(1.D0-EC**2)/(1.D0+EC*DCOS(P))          00000420
C                                           00000430
C CALCULATE UNIT VECTOR TO SUN, EARTH ECLIPTIC FRAME 00000440
C   TMP(1)=-DCOS(B)                          00000450
C   TMP(2)=-DSIN(B)                          00000460
C   TMP(3)=0.D0                               00000470
C                                           00000490
C   DO 3 I=1,3                                00000490
C   RS2(I)=0.D0                               00000500
C   DO 3 J=1,3                                00000510
C   3 RS2(I)=RS2(I)+ECLMTX(J,I)*TMP(J)        00000520
C                                           00000530
C CALCULATE UNIT VECTOR TO SUN, EQUATORIAL FRAME 00000540
C   DO 5 I=1,3                                00000550
C   RS1(I)=0.D0                               00000560
C   DO 5 J=1,3                                00000570
C   5 RS1(I)=RS1(I)+EQUMTX(J,I)*RS2(J)        00000580
C                                           09000590
C                                           00000600
C TRANSFORM TO EQUINOCTIAL COORD.            00000610

```

10
C

```
AB= 1.D0+Z(4)**2+Z(5)**2
CM(1,1)= (1-D0-2(4)**2+Z(5)**2)/AB
CM(2,1)= 2.D0*Z(4)*Z(5)/AB
CM(3,1)= -2.D0*Z(4)/AB
CM(1,2)= CM(2,1)
CM(2,2)= (1.D0+Z(4)**2-Z(5)**2)/AB
CM(3,2)= 2.D0*Z(5)/AB
CM(1,3)= -CM(3,1)
CM(2,3)= -CM(3,2)
CM(3,3)= (1.D0-Z(4)**2-Z(5)**2)/AB
DO 10 I=1,3
  RS(I)= 0.D0
DO 10 J=1,3
  RS(I)= RS(I)+CM(J,I)*RS1(J)
RETURN
END
```

```
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
```


Subroutine TRAJ (TRAJSS)

Description :

The Runge-Kutta integrator is called to calculate the low thrust trajectory. The errors in the final conditions are calculated to send back to the iterator.

Common Areas:

XMMM/ZLO(5), STEP(6), ZERF(6)

TRA/TFMAX, DT0, UEB, EW(10)

Z/Z(10), DERZ(10)

INT/IPR, IDIM, IDIM2, NIMAX

T/TF, TO

ELEM/ZP0(5), ZPF(5)

DY/DYDT(6)

TC/NOP

A/A, AMU, PI

NU/XNU

Called by :

CONTIL, ITER

Calls Subroutines :

RK4, FUNCT

C TRAJ/TRAJSS

```

C SOLAR SAIL
C INCLUDES PENALTY FUNCTION.
C THIS ROUTINE SETS UP THE INPUT TO THE INTEGRATOR WHICH
C EXTRAPOLATES THE TRAJECTORY FROM INITIAL TIME TO
C FINAL TIME. IT ALSO EVALUATES THE CHANGE IN TF AND
C THE ERROR IN THE FINAL CONDITIONS.
C THIS PROGRAM IS CALLED BY ITER OR BY CONTROL
C IT CALL THE SUBPROGRAM RK4 (RUNGA-KUTTA)
C MIN J, MAX H.
C 6 DIN. ZERF. T.C. OPTIONS.
C NOP=1--ALL 5 FINAL O.E. FIXED, =2--A,E,I ONLY FIXED.
2   =3--SEMI-MAJOR AXIS (C3) SPECIFIED, ESCAPE TRAJ
C
C SUBROUTINE TRAJ
C
C IKPLZCIT REAL*8(A-H,O-Z), INTEGER (I-N)
C
COMMON /XMMH/ZLO(5), STEP(6), ZERF(6)
COMMON /TRA/TFMAX, DTO, UEB, EW(10)
COMMON /Z/Z(10), DERZ(10)
COMMON /INT/IPR, IDIM, IDIM2, NIMAX
COMMON /T/TF, TO
COMMON /ELEM/ZP0(5), ZPF(5)
COMMON /DY/DYDT(6)
COMMON /TC/NOP
COMMON /A/A,AMU,PI
COMMON /NU/XNU
C
C EXTERNAL FUNCT, OUTP
C DIMENSION PRMT(5), AUX(1,10),DERZ1(10)
C
C
C
C IF (TF.GT.TFMAX) TF=TFMAX
C PRMT(1)= TO
C PRMT(2)= TF
C PRMT(3)= DTO
C PRMT(4)= UEB
C
C
C
C Z IS A VECTOR OF STATE AND COSTATE
C DO 10 I=1,IDIM2
C   Z(I)=ZP0(I)
10  Z(I+IDIM2)= ZLO(I)
C
C EW ARE ERROR WEIGHTS--INPUT TO THE INTEGRATOR
C
15  DO 20 I=1,IDIM
20  DERZ(I)=EW(I)
C
C CALL THE R-K INTEGRATOR
C
C CALL RK4(PRMT,Z,DERZ,IDIM,IHLF,FUNCT,OUTP,AUX)
C IF (IHLF.GT.10) GC TO 100
C
C Z IS NOX THE FINAL O.E.,
C ZERP THE ERROR IN THE FINAL CONDITIONS
C
H=0.DO

```

```

00000010
00000020
00000030
00000040
00003050
00000060
00000070
00090080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610

```

	DO 30 I=1, IDIM2	00000620
30	H= H + Z(I+5)*DERZ(I)	00030630
	TF1=TF*(STEP(6)+1.D0)	00000640
	CALL FUNCT(TF1,Z,DERZ1)	00000650
	H1=0.D0	00000660
	DO 35 I=1, IDIM2	00000670
35	H1=H1+Z(I+5)*DERZ1(I)	00000680
	DYDT(6)= (H1-H)/(TF1-TF)	00000690
C	PENALTY FUCTION ETPECT	00000700
	H= H-XNU/(2.D0*Z(1)*Z(1)*(1.D0-DSQRT(Z(2)*Z(2)*Z(3)*Z(3)))**2)	00000710
C		00000720
	ZERF(6)= H -1.D0	00000730
C		00000740
C	FINAL COYDITION OPTION BRANCH	00000750
C		00000760
	GO TO (40,50,80),NOP	00000770
C		00000780
40	DO 45 I=1,5	00000790
	ZERF(I)= Z(I) -ZPF(I)	00000800
45	DYDT(I)= DERZ(I)	00000810
	RETURN	00000820
		00000830
50	ZERF(4)= (Z(3)*Z(7)-Z(2)*Z(8))*1.D-3	00000840
	ZERF(5)= (Z(5)*Z(9) -Z(4)*Z(10))*1.D-3	00000850
	DYDT(4)= DERZ(3)*Z(7)+Z(3)*DERZ(7)-DERZ(2)*Z(8) -Z(2)*DERZ(8)	00000860
	DYDT(4)= DYDT(4)*1.D-3	00000870
	DYDT(5)= DERZ(5)*Z(9) +Z(5)*DERZ(9) -DERZ(4)*Z(10)-Z(4)*DERZ(10)	00000880
	DYDT(5)= DYDT(5)*1.D-3	00000890
C		00000900
60	ZERF(1)= Z(1) - ZPF(1)	00000910
	DUM1= DSQRT(Z(2)**2 + Z(3)**2)	00000920
	ZERF(2)= DUM1 - ZPF(2)	00000930
	DUM2= DSQRT(Z(4)**2 + Z(5)**2)	00000940
	ZERF(3)= DUM2 - ZPF(3)	00000950
	DYDT(1)= DERZ(1)	00000960
	DYDT(2)= 0.D0	00000970
	DYDT(3)= 0.D0	00000980
	IF (DUM1.GT.1.D-12) DYDT(2)= (Z(2)*DERZ(2) + Z(3)*DERZ(3))/DUM1	00000990
	IF (DUM2.GT.1.D-12) DYDT(3)= (Z(4)*DERZ(4) + Z(5)*DERZ(5))/DUM2	00001000
C		00001010
C	SPECIAL CASE, E=0 AND/OR I=0	00001020
C		00001030
	IF (ZPF(2).NE.0.D0) GO TO 70	00001040
	ZERP(2)= Z(2)	00001050
	ZERF(4)= Z(3)	00001060
	DYDT(2)= DERZ(2)	00001070
	DPDT(4)= DERZ(3)	00007080
70	IF (ZPF(3).NE.0.D0) RETURN	00001090
	ZERP(3)= Z(4)	00001100
	ZEHF(5)= Z(5)	00001110
	DYDT(3)= DERZ(4)	00001120
	DYDT(5)= DERZ(5)	00001130
	RETURN	00001140
C		00001150
80	ZERF(1)= Z(1)-ZPF(1)	00001160
	DYDT(1)= DERZ(1)	00001170
	DO 85 I=2,5	00001180
	ZERF(I)= Z(I+5)*1.D-3	00001190
85	DYDT(I)= DERZ(I+5)*1.D-3	00001200
	RETURN	00001210
C		00001220

100	IF (IHLF.EQ.11) WRITE (6,1000)	00001230
	IF (IHLF.EQ.12) WRITE (6,1001)	00001240
	IF (IHLF.EQ.13) WRITE (6,1002)	00001250
	STOP	00001260
		00001270
C		
1000	FORMAT (68H0 THE NUMBER OF BISECTIONS OF THE ORIGINAL INCREMENT HAS EXCEEDED 10)	00001280
		00031290
1001	FORMAT (27H0 INITIAL INCREMENT IS ZERO)	00001300
1002	FORMAT (54H0 INITIAL INCREMENT HAS WRONG SIGN OR BOUNDS ARE WRONG)	00001310
	END	00001320

Subroutine VCP

Description:

Calculates a vector cross product, $W = U \times V$

Argument List:

<u>U</u>	First 3-vector
<u>V</u>	Second 3-vector
<u>W</u>	Cross product result, $U \times V$

C VCP

C THIS SUBROUTINE CALCULATES THE VECTOR CROSS PRODUCT U X V
C AND RETURNS PRODUCT AS W

C

SUBROUTINE VCP (U, V, W)

C

DOUBLE PRECISION U, V, W
DIMENSION U (3), V (3), W (3)

C

W (1) = -U (3) * V (2) + U (2) * V (3)
W (2) = U (3) * V (1) - U (1) * V (3)
W (3) = -U (2) * V (1) + U (1) * V (2)
RETURN
END

00000010
00000020
00000030
00000040
00000050
09000050
00000070
00000080
00000090
00300100
09000110
00000120
00000130
00000140
00000150

LIST OF REFERENCES

1. Sackett, L. L., Optimal Solar Sail Planetocentric Trajectories, Draper Laboratory Report R-1113, September 1977.
2. Sackett, L. L., H. L. Malchow, and T. N. Edelbaum, Solar Electric Geocentric Transfer with Attitude Constraints: Analysis, NASA CR-13497, Draper Laboratory Report R-901, August 1975.
3. Sackett, L. L., H. L. Halchow, and T. N. Edelbaum, Solar Electric Geocentric Transfer with Attitude Constraints: Program Manual, Draper Laboratory Report R-902, August 1975.
4. System/360 Scientific Subroutine Package Version IV, Programmers Manual, GH20-0205-4, IBM Corp., Fifth edition, 1970.
5. Cefola, P. J., "Equinoctial Orbital Elements - Application to Artificial Satellite Orbits", AIAA Paper 72-937, Palo Alto, California, 1972.